

Marcos Antonio Andrade Silva

**Análise de desempenho do protocolo DNP3
encapsulado sobre PLC para aplicações em
Redes Inteligentes**

São Cristóvão-SE, Brasil

2017

Marcos Antonio Andrade Silva

Análise de desempenho do protocolo DNP3 encapsulado sobre PLC para aplicações em Redes Inteligentes

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Sergipe como parte do requisitos necessários para a obtenção do grau de Mestre em Engenharia Elétrica.

Área de Concentração: Telecomunicações

Universidade Federal de Sergipe – UFS

Programa de Pós-Graduação em Engenharia Elétrica – PROEE

Orientador: Prof. Dr. Leocarlos Bezerra da Silva Lima

São Cristóvão-SE, Brasil

2017



UNIVERSIDADE FEDERAL DE SERGIPE
PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA
COORDENAÇÃO DE PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA-PROEE

TERMO DE APROVAÇÃO

**“Análise de desempenho do protocolo DNP3
encapsulado sobre PLC para aplicação em Redes
Inteligentes”**

Discente:

Marcos Antônio Andrade Silva

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Sergipe, como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica.

Aprovada pela banca examinadora composta por:


Prof. Dr. Oscar Alberto Zanabria Sotomayor (PROEE/UFS)
Presidente


Prof. Dr. Ricardo José Paiva Britto Salgueiro (DCOMP/UFS)
Examinador Externo


Profa. Dra. Andréa Araújo Sousa (PROEE/UFS)
Examinador Interno


Marcos Antônio Andrade Silva
Candidato

Cidade Universitária “Prof. José Aloísio de Campos”, 28 de agosto de 2017.

**FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL
UNIVERSIDADE FEDERAL DE SERGIPE**

S586a Silva, Marcos Antonio Andrade
Análise de desempenho no protocolo DNP3 encapsulado sobre PLC para aplicações em redes inteligentes / Marcos Antonio Andrade Silva ; orientador Leocarlos Bezerra da Silva Lima. – São Cristóvão, 2017.
101 f. ; il.

Dissertação (mestrado em Engenharia Elétrica) – Universidade Federal de Sergipe, 2017.

1. Engenharia elétrica. 2. Redes elétricas inteligentes. 3. Linhas elétricas. I. Lima, Leocarlos Bezerra da Silva, orient. II. Título.

CDU: 621.3

À memória de meu pai.

Agradecimentos

Agradeço a minha família, que sem ela eu não estaria aqui hoje. A Minha mãe, que tanto batalhou para ter seus filhos formados e progredindo, minhas irmãs que sempre estiveram dispostas a me ajudar nos momentos em que foram requisitadas, especialmente Jaciara, que leu, *re-leu*, revisou, discutiu corrigiu as atividades desenvolvidas no âmbito desta dissertação e em minhas atividades desempenhadas como um todo, de todas a que tenho um maior sentimento de gratidão por toda a energia gasta para ajudar.

A minha esposa Fabíola, que ao longo do tempo passou a integrar a minha família, cuja participação foi ativa em todas as ocasiões no decorrer deste mestrado, a mudança de cidade, de emprego, de vida, uma companheira que foi e é acima de tudo uma grande amiga, sem a qual eu não teria conseguido chegar aonde eu cheguei. Obrigado pela confiança, carinho, paciência e dedicação.

Aos colegas do mestrado com os quais dividi tempos de estudos quando estávamos cursando as disciplinas, bem como nas discussões ao longo do trabalho, em especial a Guilherme, que além de haver reciprocidade na ajuda do desenvolvimento do trabalho, sempre me disse suas palavras de incentivo: “Abandone o barco!”.

Aos professores do curso que não mediram esforços para contribuir com o nosso desenvolvimento profissional, especialmente ao Professor Leocarlos Bezerra da Silva Lima pela confiança depositada em mim, por toda a ajuda, apoio, orientação e, sobretudo amizade consolidada nos trabalhos desenvolvidos.

A Oscar Jaimes, por disponibilizar seu trabalho com o protocolo DNP3 e Alcides Ortega por também ter sido extremamente solícito em ajudar, respondendo inclusive a todos os e-mails que foram enviados.

Aos antigos e novos amigos com quem tanto discuti, tirei dúvidas, fiz *brainstorming* a respeito de algum assunto, Marco Aurélio, Ronny, Rodrigo, Fagner, o grande *Mestre Jedi* e não menos importante a Rafael, Jonatas, Lahire, Emission e Elissandro, os amigos que muito incomodei no decorrer deste trabalho.

Aos colegas da Coordenação de Eletrotécnica que me apoiaram sempre que precisei especialmente, Rita, que sempre atendeu com uma relação de amizade além da relação de trabalho. Ao Instituto Federal de Educação, Ciência e Tecnologia do Sertão Pernambucano (IF Sertão-PE) e ao Programa de Pós Graduação em Engenharia Elétrica, da Universidade Federal de Sergipe – PROEE-UFS.

Enfim, a todos que direta ou indiretamente contribuíram para a conclusão deste trabalho.

“Eu sei quem eu sou e quem posso ser, se eu desejar.”
(Miguel de Cervantes)

Resumo

A utilização de redes inteligentes de energia é uma realidade em diversos países desenvolvidos, sendo uma evolução natural para a rede elétrica de qualquer país em vias de desenvolvimento, como é o caso do Brasil. Essa evolução dá-se por uma interação dos sistemas de comunicação com as redes elétricas, em uma comunicação bidirecional, tanto de dados como de fluxo de energia, utilizando para isso protocolos e tecnologias de comunicação diferentes, cuja validação ocorre com uma instalação física dos equipamentos associados, ou, como alternativa, por simulação computacional. Este trabalho demonstra através de simulações a viabilidade do uso do protocolo DNP3 (do inglês *Distributed Network Protocol*) encapsulado em TCP/IP e em PLC (do inglês *Power Line Communication*) para aplicações em redes inteligentes, exceto para atividades que necessitem de comunicação em tempo real. Foram empregados o software de simulação de redes NS-2 para modelagem do canal PLC e do protocolo DNP3 e verificados parâmetros de qualidade, como perda de pacotes, latência e vazão de dados sob diversos cenários de aplicações.

Palavras-chaves: Redes inteligentes, PLC, DNP3, NS-2.

Abstract

The use of smart grids is a reality in several developed countries, being a natural evolution for the electrical grids of any developing country, like Brazil. This evolution takes place through the interaction of communication systems with the electrical grids in a bidirectional communication of both data and energy flow using different protocols and communication technologies. Its validation occurs by the physical installation of equipments, or alternatively by computational simulators. This paper demonstrates through simulation the use viability of DNP3 (Distributed Network Protocol) encapsulated over TCP/IP over PLC (Power Line Communication). The network simulator software NS-2 was employed for PLC channel modeling and DNP3 implementation. Some quality parameters, such as packet loss, latency and throughput, were evaluated under several scenarios.

Keywords: Smart grid, PLC, DNP3, NS-2.

Lista de ilustrações

Figura 2.1 – Estrutura da rede elétrica atual.	11
Figura 2.2 – Estrutura da rede inteligente.	11
Figura 2.3 – Arquitetura <i>Smart Grid</i>	13
Figura 2.4 – Matriz Energética do Brasil – adaptada.	13
Figura 2.5 – Estrutura básica da simulação.	14
Figura 2.6 – Ligação OTcl – C++ da classe Channel.	15
Figura 2.7 – Ligação OTcl – C++.	15
Figura 2.8 – Topologia simples de rede.	16
Figura 2.9 – <i>Trace</i> de saída, resultado da simulação de rede empregando NS-2. . . .	17
Figura 2.10–Exemplo da animação do NS-2 – NAM.	18
Figura 2.11–Modulação da informação a ser transmitida.	19
Figura 2.12–Vários padrões de tecnologia PLC nos grupos NB-PLC e BB-PLC. . . .	21
Figura 2.13–Tecnologia PLC para baixa tensão.	25
Figura 2.14–Comparação entre o modelo de referência OSI e o modelo EPA.	27
Figura 2.15–Modelo mestre – escravo do protocolo DNP3.	29
Figura 2.16–Comunicação entre dispositivos.	30
Figura 2.17–Fragmentos, segmentos e quadros da mensagem DNP3.	30
Figura 2.18–Estrutura de construção de mensagem DNP3.	31
Figura 2.19–Pacotes nas camadas do protocolo DNP3.	32
Figura 2.20–Troca de datagramas no envio de um comando.	32
Figura 2.21–Topologias do protocolo DNP3.	33
Figura 2.22–Encapsulamento do protocolo DNP3 sobre TCP/IP.	35
Figura 2.23–Pilha de protocolos DNP3 sobre TCP/IP.	36
Figura 3.1 – Diagrama de estado da estação <i>Outstation</i>	38
Figura 3.2 – Diagrama de estado da estação mestre.	39
Figura 3.3 – Diagrama de classes.	40
Figura 3.4 – Ligação OTcl – C++ da classe dnp3client.	42
Figura 3.5 – Indicação do tipo de pacote PT_DNP3 no arquivo packet.h.	43
Figura 3.6 – Modificação do ns-default para o DNP3.	43
Figura 3.7 – Adicionando o DNP3 a lista de protocolos da camada de aplicação. . .	43
Figura 3.8 – Exemplo de código TCL.	44
Figura 3.9 – Gerador de tráfego e encapsulamento do agente.	44
Figura 3.10–Definição das estações.	45
Figura 3.11–Origem, destino e tamanho do pacote.	46
Figura 3.12–Ligação OTcl em channelmod.cc.	48
Figura 3.13–Trecho do código OTcl da simulação elaborada.	48

Figura 3.14–Cálculo do PER de acordo com a BER.	49
Figura 3.15–Taxonomia dos protocolos de acesso múltiplo.	51
Figura 3.16–Declaração de Métodos na classe MAC.	52
Figura 3.17–Diagrama de classes da Subcamada Mac.	53
Figura 3.18–Ponto a ponto – validação.	54
Figura 3.19–Mensagem não solicitada e função de leitura.	54
Figura 3.20–Multiponto – validação.	55
Figura 3.21–Simulação Multiponto.	55
Figura 4.1 – Trecho do código OTcl para simulação do DNP3.	57
Figura 4.2 – Mensagem de saída no <i>prompt</i> de comando.	57
Figura 4.3 – Arquivo na simulação do DNP3.	58
Figura 4.4 – Mensagem de saída da função <i>reset</i> no <i>prompt</i> de comando.	59
Figura 4.5 – Interligação da estação remota no consumidor à mestre na rede de distribuição.	61
Figura 4.6 – Topologia do cenário de simulação da comunicação ponto a ponto. . . .	62
Figura 4.7 – Taxa de Perdas x SNR – Modulação DQPSK e DBPSK.	65
Figura 4.8 – Vazão média x SNR – Modulação DQPSK e DBPSK.	66
Figura 4.9 – Perda de Pacotes x Ocupação do canal – Modulação DQPSK e DBPSK. .	66
Figura 4.10–Vazão média x Ocupação do canal – Modulação DQPSK e DBPSK. . .	67
Figura 4.11–Latência x Ocupação do canal – Modulação DQPSK e DBPSK.	67
Figura 4.12–Interligação das estações remotas no consumidor à mestre na rede de distribuição.	69
Figura 4.13–Animação NAM da configuração multiponto.	70
Figura 4.14–Perda de pacotes x SNR.	72
Figura 4.15–Quantidade de pacotes com latência acima da média – DQPSK.	73
Figura 4.16–Quantidade de pacotes com latência acima da média – DBPSK.	74
Figura 4.17–Vazão de dados – Modulação DQPSK e DBPSK.	74
Figura 4.18–Animação NAM da configuração multiponto com tráfego heterogêneo. .	76
Figura 4.19–Ocupação do canal na rede com tráfego heterogêneo.	76
Figura 4.20–Latência de dados x Taxa de Ocupação – rede de tráfego heterogêneo. .	77
Figura 4.21–Latência de dados em rede com tráfego heterogêneo.	77
Figura 4.22–Vazão de dados em rede com tráfego heterogêneo.	78

Lista de tabelas

Tabela 2.1 – Regulação da Compatibilidade Eletromagnética – EMC para BB e NB-PLC.	22
Tabela 2.2 – Faixas de exclusão para uso do PLC no Brasil.	22
Tabela 2.3 – Faixas de Radiofrequência relativas à zona de proteção de estações costeiras.	23
Tabela 2.4 – Características das camadas DNP3.	28
Tabela 2.5 – Níveis de implementação do protocolo DNP3.	35
Tabela 3.1 – Métodos das classes <code>dnp3AppMaster</code> e <code>dnp3AppClient</code>	41
Tabela 3.2 – Relação entre SNR e BER na modulação DQPSK.	48
Tabela 3.3 – Relação entre SNR e BER na modulação DBPSK.	49
Tabela 4.1 – Envio das funções básicas do DNP3.	58
Tabela 4.2 – Resultado do envio cíclico das mensagens não solicitadas.	59
Tabela 4.3 – Requerimento de Qualidade de Serviço (QoS) e tipos de tráfego em rede de comunicação inteligente.	60
Tabela 4.4 – Envio de mensagens não solicitada na comunicação ponto a ponto.	62
Tabela 4.5 – Envio de mensagens de leitura na comunicação ponto a ponto.	64
Tabela 4.6 – Envio de mensagens de escrita na comunicação ponto a ponto.	64
Tabela 4.7 – Parâmetros iniciais da análise estatística da simulação.	68
Tabela 4.8 – Parâmetros iniciais da análise estatística da simulação.	69
Tabela 4.9 – Parâmetros de entrada da simulação multiponto.	71
Tabela 4.10–Parâmetros de entrada – outras funções.	73
Tabela 4.11–Número de pacotes enviados na simulação multiponto – DQPSK.	75
Tabela 4.12–Envio de pacotes em rede com tráfego heterogêneo.	78

Lista de abreviaturas e siglas

ABNT	Associação Brasileira de Normas Técnicas
ACK	<i>Acknowledgement</i> Confirmação
ANATEL	Agência Nacional de Telecomunicações
ANEEL	Agência Nacional de Energia Elétrica
APCI	<i>Application Protocol Control Information</i> – Controle de Informação de Protocolo de Aplicação
APDU	<i>Application Protocol Data Unit</i> – Unidade de Dados de Protocolo de Aplicação
ASDU	<i>Application Service Data Unit</i> – Unidade de Dados de Serviço de Aplicação
BCD	Decimal Codificado em Binário
BER	<i>Bit Error Rate</i> – Taxa de Erro de Bit
CBR	<i>Constant Bit Rate</i> – Taxa de Bit Constante
CEMIG	Companhia Energética de Minas Gerais
COPEL	Companhia Paranaense de Energia
CRC	<i>Cyclic Redundancy Check</i> – Verificação Cíclica de Redundância
CSMA	<i>Carrier Sense Multiple Access</i> – Acesso Múltiplo de Detecção de Portadora
CSMA/CA	<i>Carrier Sense Multiple Access with Collision Avoidance</i> – Acesso Múltiplo de Detecção de Portadora com Prevenção de Colisão
DBPSK	<i>Differential Binary Phase-Shift Key</i> – Modulação Diferencial Binária por deslocamento de fase
DHT	Distorção Harmônica Total
DNP3	<i>Distributed Network Protocol</i> – Protocolo de Rede Distribuída versão 3
DQPSK	<i>Differential Quadrature Phase-Shift Key</i> – Modulação Diferencial por Chaveamento de Fase e Quadratura

EPA	<i>Enhanced Performance Architecture</i> – Arquitetura de Desempenho Aprimorada
FTP	<i>File Transfer Protocol</i> – Protocolo de Transferência de Arquivos
IEC	<i>International Electrotechnical Commission</i> – Comissão Eletrotécnica Internacional
IED	<i>Intelligent Electronic Device</i> – Dispositivo Eletrônico Inteligente
IEEE	<i>Institute of Electrical and Electronics Engineers</i> – Instituto de Engenharia Elétrica e Eletrônica
IP	<i>Internet Protocol</i> – Protocolo de Internet
ISO	<i>International Organization for Standardization</i> – Organização Internacional de Padronização
LPDU	<i>Link Protocol Data Unit</i> – Unidade de Dados de Protocolo de Enlace
MAC	<i>Media Access Control</i> – Controle de Acesso ao Meio
NAM	<i>Network Animator</i> – Animador de Rede
NS-2	<i>Network Simulator version 2</i> – Simulador de Redes versão 2
OSI	<i>Open System Interconnection</i> – Interconexão de Sistemas Abertos
OTcl	<i>Object-oriented Tool Command Language</i> – Linguagem de Comando de Ferramentas Orientada a objetos
PER	<i>Packet Error Rate</i> – Taxa de Erro de Pacotes
PLC	<i>Power Line Communication</i> – Comunicação na Rede Elétrica
RTU	<i>Remote Terminal Unit</i> – Unidade Terminal Remoto
SCADA	<i>Supervisory Control and Data Acquisition</i> – Sistema de Supervisão e Aquisição de Dados
SNR	<i>Signal Noise Rate</i> – Relação Sinal Ruído
TCP	<i>Transmission Control Protocol</i> – Protocolo de Controle de Transmissão
TPDU	<i>Transport Protocol Data Unit</i> – Unidade de Dados de Protocolo de Transporte
TSDU	<i>Transport Service Data Unit</i> – Unidade de Dados de Serviço de Transporte

UDP	<i>User Datagram Protocol</i> – Protocolo de Datagrama de Usuário
VLAN	Rede Virtual
VoIP	Voz sob IP

Sumário

1	INTRODUÇÃO	1
1.1	Motivação	2
1.2	Objetivo	2
1.2.1	Geral	2
1.2.2	Específicos	3
1.3	Metodologia	3
1.4	Trabalhos Relacionados	4
1.4.1	PLC	4
1.4.2	DNP3	7
1.4.3	NS-2	8
1.5	Escopo do trabalho	9
2	FUNDAMENTAÇÃO TEÓRICA	10
2.1	Redes Inteligentes	10
2.1.1	Sistema tradicional x redes inteligentes	11
2.1.1.1	Fluxo de energia	11
2.1.1.2	Fluxo de dados	12
2.1.1.3	Variedade de formas de geração	12
2.2	NS-2	14
2.2.1	A simulação no NS-2	14
2.2.2	Arquivo trace de saída	16
2.2.3	Objeto animado NAM	17
2.3	PLC	18
2.3.1	Processo de modulação	19
2.3.2	Frequência de operação	20
2.3.2.1	Padrões, Especificações e Regulamentação de Frequência	20
2.3.3	PLC em Redes Inteligentes	21
2.3.3.1	PLC em redes de média tensão	24
2.3.3.2	PLC em redes de baixa tensão	25
2.4	DNP3	26
2.4.1	Recursos do protocolo	26
2.4.2	Estrutura do protocolo	27
2.4.3	Dispositivos	28
2.4.4	Construção e troca de mensagens	29
2.4.5	Topologias	33

2.4.6	Integração com o TCP/IP	34
3	DESENVOLVIMENTO DO TRABALHO	37
3.1	Utilização do DNP3 no NS-2	37
3.1.1	Estrutura de dados do protocolo	37
3.1.2	Ligação OTcl – C++	42
3.1.3	Alteração de arquivos do núcleo do programa	43
3.1.4	O código TCL	44
3.2	Implementação do PLC	45
3.2.1	Modelagem do Canal	46
3.2.2	Controle de Acesso ao Meio	50
3.2.2.1	Comunicação ponto a ponto	51
3.2.2.2	Comunicação multiponto	52
3.2.3	Validação da implementação	54
4	SIMULAÇÕES E RESULTADOS	56
4.1	Simulação do DNP3	56
4.1.1	Envio das funções básicas do DNP3	56
4.2	Simulação do DNP3 encapsulado em PLC	60
4.2.1	Comunicação ponto a ponto para envio de dados DNP3	61
4.2.1.1	Envio de uma mensagem não solicitada pela estação remota	62
4.2.1.2	Solicitação de leitura de dados feita pelo mestre	63
4.2.1.3	Envio de dados a serem escritos na estação remota	64
4.2.1.4	Envio cíclico de mensagens não solicitadas	65
4.2.1.5	Intervalo de confiança	68
4.2.2	Comunicação multiponto para envio de dados DNP3	68
4.2.2.1	Envio de mensagem não solicitada com rede dedicada	70
4.2.2.2	Envio de outras funções junto ao envio de mensagens não solicitadas	72
4.2.2.3	Envio de mensagem não solicitada com rede de dados heterogêneos	75
5	CONCLUSÕES	79
	REFERÊNCIAS	81

1 Introdução

Tornar o consumo de energia mais eficiente, ter um fornecimento de energia mais transparente ao usuário, além de integrar fontes de geração distribuída¹ e aumentar a participação de fontes renováveis na matriz energética, são algumas das aplicações de uma rede inteligente ou *smart grid*. Esse tipo de rede elétrica já é realidade em alguns países da Europa, da Ásia e nos Estados Unidos, estando ainda em desenvolvimento no Brasil.

Esse desenvolvimento está ocorrendo na área acadêmica com a realização de pesquisas, pelas concessionárias de energia na elaboração de projetos pilotos, formando as chamadas *smart cities* – cidades inteligentes, que consistem na utilização dessa tecnologia visando à habilitação de diversos serviços de utilidade pública, além dos serviços de distribuição de eletricidade. Atualmente existem alguns projetos piloto em andamento, em 7 Estados diferentes, como o *Smart Grid Light* no Rio de Janeiro um dos maiores existentes em consumidores atingidos.

Essas ações podem ser vistas também por parte dos agentes reguladores, com a aprovação das resoluções 482/2012 e 687/2015 da Agência Nacional de Energia Elétrica – ANEEL que normatizam o uso de micro e mini geração por parte dos consumidores de energia e a ligação dessas à rede elétrica. Pode-se dizer que a ANEEL com seu Projeto Estratégico de Pesquisa e Desenvolvimento ajudou no crescimento dos estudos no país, principalmente por parte das concessionárias de energia elétrica.

Para implantar uma rede inteligente, deve ocorrer uma ligação harmoniosa entre a rede elétrica e os sistemas de comunicações, independentemente da tecnologia utilizada, como Ethernet, muito empregada para interconexão de redes locais, ZigBee, uma forma de comunicação sem fio com baixo consumo de energia, ou PLC – *Power Line Communication* (Comunicação na Rede Elétrica), que faz uso dos condutores já existentes na rede elétrica, modulando um sinal de alta frequência² sob a frequência fundamental de 60Hz.

Essas formas de comunicação são monitoradas e controladas por meio de protocolos de comunicação, que podem ser do tipo proprietário (privados) ou públicos. São exemplos os protocolos X-10 e o HomePlug utilizados em automação residencial, bem como IEC 61850 e DNP3 – *Distributed Network Protocol* versão 3 (Protocolo de Distribuição de Rede versão 3), muito usados na automação de subestações (ORTEGA; SHINODA, 2013).

O DNP3 tem sido utilizado na maior parte dos projetos pilotos em andamento no Brasil e, segundo Ramalho et al. (2013), apresenta como vantagens a confiabilidade,

¹ Consiste na micro ou minigeração distribuída, que são centrais geradoras provenientes dos consumidores com potência instalada de até 5MW, injetando energia elétrica na rede (ANEEL, 2015).

² Aprovada pela resolução 527 da Agência Nacional de Telecomunicações – ANATEL, a faixa de frequência permitida para uso do PLC é de 1,705 a 50 MHz.

escalabilidade e interoperabilidade de equipamentos em aplicações de redes inteligentes. Shin et al. (2015) apud CLARKE; REYNOLDS, 2004) afirmam que o DNP3 é o padrão mais importante em sistemas SCADA – *Supervisory Control and Data Acquisition* (Sistema de Supervisão e Aquisição de Dados) para sistemas de potência, sendo necessário para a integração de métodos num ambiente de rede inteligente. Pereira (2015) diz ainda que serviços oferecidos como fragmentação dos dados, verificação de erros e controle de enlace tornam-no robusto e eficiente.

Contudo, para utilização desses protocolos na integração e validação desses sistemas, são necessários testes e projetos avançados (PEREIRA, 2015), o que torna muitas vezes o processo oneroso, tendo como solução abordada por Ortega e Shinoda (2013) o uso de simuladores, que permitem a montagem de vários cenários para obtenção de resultados, assemelhando-se aos testes executados em laboratórios em pequena escala ou testes de campo que possuem como revés um alto custo. Para realizar simulações, diversas plataformas podem ser utilizadas, como o OPNET, CloMoSim, NCTUns, sendo destacado por Santos (2013) o uso do *Network Simulator Version 2* – NS-2 por ser um simulador de sistemas a eventos discretos, de código aberto, com flexibilidade e modularidade que permite validar o desempenho de um protocolo de comunicação em diferentes tipos de redes.

1.1 Motivação

ANEEL (2010a) diz que os consumidores têm exigido cada vez mais confiabilidade e qualidade da energia elétrica, exigência essa que tornou-se mais acentuada após o *blecaute* ocorrido em novembro de 2009. A ideia é que com a implantação de uma rede inteligente, a comunicação entre todas as etapas do Sistema Elétrico de Potência tornaria-se mais eficiente, diminuindo a assimetria das informações, inclusive por parte do consumidor.

Essa comunicação têm sido realizada com diversos tipos de tecnologias, sendo o PLC uma forma de comunicação madura, robusta e confiável (COELHO, 2015), que associada ao DNP3, um protocolo também maduro, apresenta perspectivas bastante favoráveis para sua utilização em redes inteligentes. Ao associar a implementação dessas tecnologias com um simulador gratuito e consolidado, a viabilidade do trabalho torna-se ainda mais acentuada, devido ao baixo custo de implementação.

1.2 Objetivo

1.2.1 Geral

O objetivo geral desse trabalho é propor e analisar a utilização do protocolo DNP3 encapsulados sobre TCP/IP sobre PLC para aplicações de redes inteligentes.

1.2.2 Específicos

- Verificar se parâmetros de desempenho de rede, como taxa de perda de pacotes, *throughput*, *jitter* e latência, atendem aos requisitos de aplicação em redes inteligentes;
- Publicar os resultados desse trabalho em periódicos e eventos científicos.

1.3 Metodologia

O desenvolvimento desse trabalho teve início com um levantamento bibliográfico amplo, que objetivou posicionar o aqui proposto no contexto do estado da arte atual. Um estudo aprofundado do NS-2 foi efetuado visando uma implementação de redes de comunicação e simulações de protocolos. Para isso, primeiramente foi feita a instalação da versão 2.35 do software. Apesar do NS-2 ser um software descontinuado, tendo o NS-3 como substituto natural, a sua escolha como ferramenta metodológica se deu em virtude da utilização do trabalho já implementado por [Jaimes \(2012\)](#), do padrão DNP3. Concomitantemente foram feitos outros dois estudos, as aplicações das tecnologias PLC e o padrão DNP3:

- a) **Tecnologias PLC:** Estudados os padrões, frequências utilizadas, esquemas de modulação, entre outros parâmetros.
- b) **Padrão DNP3:** Apresentado em [IEEE \(2012\)](#), foram estudadas suas características, especificações técnicas e camadas de utilização do modelo de referência OSI³.

A fundamentação teórica, além dos trabalhos relevantes citados na Seção 1.4, pode ser obtida em referências sobre os tópicos abordados, tais como:

- [Clarke e Reynolds \(2004\)](#).
- [Ferreira e Lampe \(2010\)](#).
- [Toledo \(2012\)](#).
- [Bush \(2014\)](#).
- [Sato et al. \(2015\)](#).

Com a devida base teórica, foram feitas algumas simulações de redes no NS-2, com as classes disponíveis no programa, analisando os resultados apresentados nos arquivos de saída, para com isso simular o protocolo DNP3.

³ Elaborado no final dos anos 1970, teve como objetivo organizar as redes de computadores em sete camadas, denominadas modelo de Interconexão de Sistemas Abertos – *Open Systems Interconnection* ([KUROSE; ROSS, 2013](#)).

A simulação do DNP3 no NS-2 foi implementado por [Jaimes \(2012\)](#) e adaptado para utilização neste trabalho. A utilização contou com a modificação e adição de classes no núcleo do NS-2 e elaboração de arquivo tcl para criação de estações e indicação dos cenários a serem simulados.

Após o devido funcionamento do DNP3 no NS-2, foi feita a implementação do PLC, realizando a modelagem do canal e o controle de acesso ao meio para tratamento do congestionamento. Com o PLC em funcionamento foi possível simular nele o envio de pacotes DNP3 e com isso coletar resultados para validação do sistema.

Essa validação se deu com o tratamento dos arquivos de saída (*trace*), animando com auxílio da ferramenta de animação de rede do software *Network Animator*⁴, do envio de pacotes DNP3 sob PLC, em diferentes cenários, sendo por fim, feita uma análise comparativa do sistema proposto com outras tecnologias, bem como a análise de parâmetros de desempenho.

1.4 Trabalhos Relacionados

A implantação de uma rede inteligente dá-se associada a uma diversidade de fatores como o aumento da representatividade na matriz energética de fontes de energia renováveis, a inserção de fontes de geração distribuída de energia, ou ainda a diminuição dos furtos de energia ([PEREIRA, 2015](#)). A motivação decorrente desses fatores impulsiona a realização de trabalhos acadêmicos em várias subáreas, sendo uma delas a área de comunicações. Nesta, a utilização de PLC e do protocolo DNP3 têm sido estudados, algumas vezes em ambientes simulados. Aqui é apresentada uma revisão bibliográfica com trabalhos relacionados a este.

1.4.1 PLC

[Santos et al. \(2009\)](#) modelaram o tráfego VoIP (Voz sobre IP) e transferência de dados utilizando redes PLC nas larguras de faixa entre 4,3 e 20,9MHz para transmissão. Essa modelagem foi feita baseada nas Cadeias de Markov, sendo realizadas simulações laboratoriais em ambientes *indoor*, tendo como conclusão a eficiência do uso do modelo MMFM (*Markov Modulated Fluid Models*) para tráfegos reais de VoIP e de dados da rede PLC em detrimento de outros modelos Markovianos testados.

[Dias et al. \(2010\)](#) apresentaram uma modelagem de conexão de um sistema PLC *outdoor* na rede de distribuição de energia elétrica por meio de simulação utilizando o *software* Matlab/Simulink, com o objetivo de verificar seu desempenho. Na simulação foi

⁴ *Network Animator*, ou simplesmente NAM, é uma ferramenta de animação baseada em linguagem Tcl, para visualização de rastreamento de pacotes em redes simuladas e em redes do mundo real ([INSTITUTE, 2016](#)).

utilizado o limite de frequência inferior permitido pela ANATEL de 1,705 MHz e a rede elétrica considerada como um canal de multi-propagação devido às reflexões geradas pelos multi percursos com descontinuidades de impedância. O trabalho, porém, não contemplou a presença dos ruídos proveniente dos chaveamentos, interferências eletromagnéticas, sobretensões ou correntes de curto-circuito, sugerindo como um possível trabalho futuro.

Kellerbauer e Hirsch (2011) propuseram a simulação para acesso e aplicações *inhouse* (indoor) em um cenário de rede inteligente, com base nos projetos Eenergy⁵ e E-DeMa⁶. Tal simulação é implementada em C++ *builder* usando OMNeT++, uma ferramenta de simulação baseada em eventos de rede e seu submódulo INET, uma coleção de unidades e tecnologias comuns de rede.

Meftah et al. (2011) apresentaram uma simulação da tecnologia PLC utilizando o padrão HomePlug AV – HPAV, tendo como software de simulação o NS-2. No trabalho é mostrado que o modelo simulado, que reproduz o desempenho observado nos equipamentos HomePlug, requer o desenvolvimento de um mecanismo planejador eficiente para o acesso CSMA⁷. Além disso, a investigação do processo de segmentação leva a conclusões sobre o impacto deste processo no desempenho do TCP.

Augusto (2012) apresentou um modelo de simulação para PLC, operando em um ambiente de redes inteligentes, em que foi utilizado um modelo de propagação multi percursos e considerado o efeito do atraso de propagação do canal. No trabalho também foi proposto um algoritmo de adaptação de taxa de dados, baseado no utilizado em tecnologias sem fios. Esse trabalho foi baseado no modelo básico proposto por Kellerbauer e Hirsch (2011), porém oferecendo uma camada física mais realista, que apresentou um pequeno custo em termos de desempenho.

Já Schreiber (2013) fez a modelagem matemática de um segmento de rede de distribuição elétrica validando a utilização do modelo PI para representar a transmissão de um sinal PLC em uma rede elétrica de baixa tensão. Na validação foi realizada a construção em pequena escala de uma rede de distribuição e posteriormente a simulação utilizando o ambiente Matlab/Simulink, na qual foram modulados sinais na frequência de 90kHz e testada a atenuação do sinal modulado, apresentando resultados pouco superiores a 60%, porém dentro dos limites de amplitude que podem ser recebidos por Modems PLC.

Mathias (2013) fez a extração de parâmetros característicos das instalações brasileiras para modelagem de canal PLC de aplicação *indoor*, simulando e utilizando o protocolo KNX na comunicação dos dispositivos de aplicação de automação residencial.

⁵ Programa que quando em funcionamento buscou formas de reduzir o consumo de energia, a efficientização do seu uso e incorporação de fontes de energias sustentáveis.

⁶ Programa também expirado do governo alemão em que eram realizados testes de campo para o consumo inteligente de energia elétrica em casas particulares.

⁷ *Carrier Sense Multiple Access* – Acesso múltiplo com sensoramento da portadora, é um protocolo de controle de acesso ao meio que busca evitar a colisão de pacotes em redes com múltiplo acesso ao meio

Nele, foi feita a comparação de modelos de resposta do canal PLC, escolhendo o modelo que apresentou maior flexibilidade a alteração de percursos. O trabalho concluiu também que a modelagem do canal como um modelo LPTV (*Linear Periodically Time-Varying*) apresentou resultados mais realistas do que o LIT (Linear Invariante no Tempo).

Lobato (2014) fez a aplicação em barramentos de corrente contínua de sinais PLC para aplicação em um ambiente industrial, visando avaliar a utilização do PLC como forma de transmissão das informações necessárias ao comando e leitura de parâmetros de acionamentos de motores elétricos baseados em conversores de frequência, utilizando como meio de comunicação o próprio barramento de corrente contínua do conversor. Neste trabalho ele concluiu como factível essa forma de transmissão de dados, inclusive em ambientes onde o inversor/motor ficam em locais inacessíveis.

Bongiolo e Miranda (2014) analisaram a situação de regulação de redes inteligentes na Europa e o modo de realização da comunicação dentro dessa nova rede com foco na tecnologia PLC. Foi realizada a montagem de um sistema para verificação de alguns casos de uso como gerenciamento de apagões, proteção de recursos da rede, entre outros. Nesse sistema, gerenciado por meio de um supervisório implementado em *Lab View*, foram realizados testes de sobrecarga e de proibição de acionamento da fonte de geração distribuída, sendo a comunicação realizada por meio da rede elétrica. Por fim eles reforçaram o uso da tecnologia PLC como uma alternativa para transmissão de dados, indicando como uma das tecnologias mais viáveis para instalação a curto prazo.

Coelho (2015) analisou o monitoramento do consumo de energia utilizando PLC, no qual foi implantado e validado em uma aplicação *indoor* o uso da tecnologia PLC em uma residência. Essa implementação se deu com a elaboração de um protótipo controlado por um microcontrolador em um sistema do tipo mestre-escravo, em que foi possível realizar o acionamento de dispositivos, controle de carga, medição de tensão, corrente, proteção contra sobrecargas e principalmente o controle de consumo de energia, tudo de forma transparente ao usuário, sendo o protocolo de comunicação o KNX, protocolo de automação residencial que permite a interoperabilidade dos equipamentos e, por consequência, a utilização de outros equipamentos já disponíveis no mercado.

Pilarski (2015) fez uma revisão sobre a tecnologia PLC, apresentando as características, vantagens e desvantagens sobre outras tecnologias de comunicação com vistas a inclusão digital no Brasil. Foram apresentadas as faixas de exclusão para uso da tecnologia PLC que, apesar de estarem dentro da faixa de frequência liberada pela ANATEL, são utilizadas por Radioamadores e Serviço Móvel Aeronáutico. No trabalho foram indicadas ainda as aplicações dessa tecnologia como rede doméstica, VLAN, VoIP, monitoramento e vigilância, telemetria. Por fim, foram apresentados dois estudos de caso de aplicação de PLC por concessionárias de energia, tais como a Companhia Paranaense de Energia – COPEL e a Companhia Energética de Minas Gerais – CEMIG.

1.4.2 DNP3

Mohagheghi et al. (2009) apresentaram uma visão geral do estado atual das redes de comunicação para subestações usando IEC 61850, e também discutiram as possíveis tendências futuras para alargar o âmbito da norma e utilizar as suas capacidades para outras aplicações dentro do sistema de distribuição. No trabalho foi feita ainda uma abordagem geral sobre protocolos de redes industriais, como o Modbus, DNP3 e IEC 61870-5, fazendo inclusive a comparação entre eles e o IEC 61850.

Ramalho et al. (2013) fizeram a modelagem de máquinas de estado para o encapsulamento do protocolo DNP3 em rede Ponto a Ponto ZigBee com o objetivo de fundir as características dos protocolos DNP3 e IEEE 802.15.4, apresentando uma solução que fornece os benefícios de ambiente sem fio, com a segurança necessária de uma aplicação de redes inteligentes. O trabalho foi modelado com a linguagem de descrição de *hardware* VHDL e propôs o monitoramento de dados residenciais por meio dos consumidores, permitindo-os a realização de controle de vários pontos de sua rede interna de energia, além da interação com aparelhos eletrodomésticos. Por fim, indicou o uso da mesma interface para transmissões de dados para um Medidor Inteligente de Vizinhaça (*Neighborhood Area Network* – NAN).

Zambenedetti et al. (2013) desenvolveram seu trabalho objetivando a criação de um novo sistema de comunicação com alta permeabilidade, capacidade de expansão, confiabilidade e de aplicação apropriada para redes de distribuição de energia elétrica. O trabalho visou ainda a implementação de monitoração, medição e controle dos dispositivos disponíveis na rede elétrica, sendo realizada também a instalação de campo de uma rede em malha, sem fio, em que os dispositivos finais e *gateways* utilizados fossem compatíveis com o protocolo DNP3, cobrindo a rede de distribuição de energia da cidade de Niterói e a monitoração e controle da iluminação pública na cidade de Curitiba.

Bani-Ahmed et al. (2014) fizeram uma análise dos desenvolvimentos tecnológicos relacionados com protocolos e padrões de sistema de comunicação de microrredes, tendo em vista que arquitetura, protocolos e ferramentas adequados de sistema de comunicação eram essenciais para garantir um funcionamento estável, confiável e ideal de uma microrrede. Eles analisaram também as camadas físicas aplicáveis às comunicações de microrredes, discutindo também as pesquisas relacionadas a área da comunicação.

Shin et al. (2015) propuseram um *gateway* M2M – *machine-to-machine* baseada em CoAP – *Constrained Application Protocol* para um sistema de automação da distribuição de energia usando DNP3.0 em um ambiente de redes inteligentes. No trabalho eles utilizaram o OPNET *Modeler* 17,1 para demonstrar as vantagens do protocolo CoAP em comunicação M2M, identificando inclusive que o CoAP tem pacotes mais rápidos e menores, comparado com o SOAP – *Simple Object Access Protocol*, sendo melhor em ambientes M2M porque a

capacidade dos dispositivos pode ser limitada.

Darwish et al. (2015) fizeram uma importante análise quanto às diferentes ameaças à segurança e vulnerabilidades a que uma rede inteligente está sujeita na utilização do DNP3 como um protocolo de comunicação em tempo real. No trabalho dois cenários de ataques foram verificados experimentalmente, um ataque de mensagem não solicitada e de injeção constante de dados. Os experimentos foram executados em um ambiente virtual de computador e, em seguida, simulados em uma plataforma de teste DETER. Nele ainda foram usadas técnicas de mitigação para garantir uma verificação saudável da rede, propondo o uso de agente de detecção de intrusão baseado em *host* em cada IED – *Intelligent Electronic Device* (Dispositivo eletrônico inteligente) com a finalidade de detectar a intrusão e mitigá-la.

1.4.3 NS-2

Xu e Yang (2011) conceberam um modelo de rede sem fio de quatro nós com base na plataforma de simulação NS-2. Nele, foram simulados o desempenho da rede, tais como atraso do sistema, a taxa de perda de pacotes, throughput e outros indicadores-chave. Ao correr dos protocolos TCP e UDP, foi feita a análise e avaliação de desempenho da rede a partir de diferentes aspectos, e, em seguida, os parâmetros associados comparados graficamente com ferramentas gnuplot, tendo os resultados da simulação e conclusões um valor positivo para a criação da topologia da rede sem fios e a configuração dos elementos de rede objeto de estudo.

Ortega e Shinoda (2013) simularam no NS-2 o protocolo DNP3 encapsulado sob TCP/IP em aplicações de redes inteligentes, realizando uma análise do comportamento de uma rede de dimensões médias ou grandes. Na simulação foram utilizadas funções básicas, como leitura de objetos de dados, das saídas digitais/analógicas, operações de controle de saída em uma rede LAN cabeada.

Pereira (2015) propôs a simulação no NS-2 do protocolo DNP3 sobre o protocolo de redes sem fio IEEE 802.15.4 para simulação de baixo custo para aplicações de redes inteligentes. No trabalho a autora avaliou cenários ponto-a-ponto e multiponto e indica o uso do mecanismo proposto para simular outros cenários de aplicações de redes inteligentes que demandem o uso da tecnologia sem fio. Ela confirmou também a eficientização quanto ao uso do NS-2 como ferramenta para análise de desempenho de sistemas, tendo em vista que os modelos presentes no simulador permitem a simulação de cenários para muitas das redes atuais em um ambiente seguro e com baixo custo.

Kumar et al. (2015) desenvolveram e validaram um protocolo NS-2 para segurança de dados na camada de rede. No trabalho foi mostrado que, apesar do NS-2 ser uma ferramenta de código aberto muito popular para simulação de rede, os protocolos nele

implementados não fornecem funções de segurança, como as exigidas em algumas aplicações como encriptação / desencriptação e troca de chaves, o que justifica a adição de um novo módulo de segurança ou protocolo em NS2, detalhando a simulação do protocolo de compartilhamento de chave e auto-definição do protocolo de encriptação / desencriptação em rede com fios e apresentando por fim essa simulação no ambiente animado NAM.

1.5 Escopo do trabalho

Essa dissertação está dividida em cinco capítulos:

No Capítulo 1 é apresentada a introdução desse trabalho, com seus objetivos, metodologia empregada, bem como a revisão bibliográfica dos trabalhos relacionados ao proposto nessa dissertação.

No Capítulo 2 é realizada a fundamentação teórica do trabalho, detalhando o conceito de redes inteligentes e descrevendo o NS-2, PLC e DNP3.

O Capítulo 3 apresenta o desenvolvimento do trabalho, descrevendo a utilização do DNP3, a implementação do PLC e sua utilização conjunta (DNP3 e PLC) no NS-2.

No Capítulo 4 são mostradas as simulações e a análise dos resultados obtidos.

No Capítulo 5 são feitas considerações finais e sugestões para continuação futura desse trabalho.

2 Fundamentação Teórica

Em uma rede inteligente há o fluxo conjunto de dados e energia e sua implantação está associada a uma diversidade de fatores, como o aumento da participação de fontes de energia renováveis na matriz energética, a inserção de fontes de geração distribuída de energia, ou ainda a diminuição dos furtos de energia (PEREIRA, 2015).

A motivação decorrente desses fatores impulsiona a realização de trabalhos acadêmicos em várias subáreas, sendo uma delas a área de comunicações, fazendo o uso de tecnologias como Ethernet, muito empregada para interconexão de redes locais, *ZigBee*, uma forma de comunicação sem fio com baixo consumo de energia, ou PLC, que modula um sinal de alta frequência nos condutores da rede elétrica já existente.

Esses trabalhos são muitas vezes realizados em ambientes simulados, com simuladores já conceituados como o MATLAB/SIMULINK, SCILAB, OCTAVE, que são softwares matemáticos, bem como NS-2, NS-3 e OMNET++, simuladores de redes, sendo o uso de simuladores uma opção que normalmente possui o melhor custo-benefício, tendo em vista o custo da implementação física dos sistemas pesquisados.

O presente Capítulo tem por objetivo fundamentar o uso do software de simulação escolhido para o desenvolvimento deste trabalho, bem como de outros conceitos cujo embasamento foi necessário para sua elaboração.

2.1 Redes Inteligentes

O Sistema Elétrico de Potência existente é um sistema unidirecional, isto é, seu sentido parte unicamente da geração até o consumidor final. Com a regulamentação da geração distribuída, começa-se a mudar esse panorama, fazendo com que o consumidor opere também como fonte geradora de energia, alimentando o sistema. Essa mudança é uma das premissas do conceito de redes inteligentes.

Segundo Fang et al. (2012) uma rede inteligente pode ser definida como: “Um sistema dotado de fluxos bidirecionais de eletricidade e informações para criar uma rede de distribuição de energia avançada, automatizada e efetivamente distribuída”. Leite (2013) afirma que uma rede passa a ser inteligente quando é constituída de modo a melhorar a prestação dos serviços de eletricidade com foco na solução de demandas sociais, políticas, econômicas, extrapolando o próprio setor elétrico.

2.1.1 Sistema tradicional x redes inteligentes

Na análise do sistema elétrico nos Estados Unidos, [Vondrasek \(2011\)](#) indica como sendo três as principais diferenças entre o sistema elétrico existente e utilizando redes inteligentes, sendo eles: o fluxo de energia, fluxo de dados e variedade de formas de geração.

2.1.1.1 Fluxo de energia

O sistema tradicional apresenta fluxo unidirecional, como pode ser visto na Figura 2.1. Nele a energia parte da geração, passando pela transmissão e por conseguinte a distribuição e o consumidor final.



Figura 2.1 – Estrutura da rede elétrica atual.

Já na rede inteligente, o fluxo de energia forma um ciclo que é reforçado com a participação dos consumidores que atuam também como fornecedores de energia, como pode ser visto na Figura 2.2. Uma das vantagens do uso das redes inteligentes, é que o consumidor tem a capacidade de se juntar às fontes geradoras de energia, reforçando assim o fornecimento.

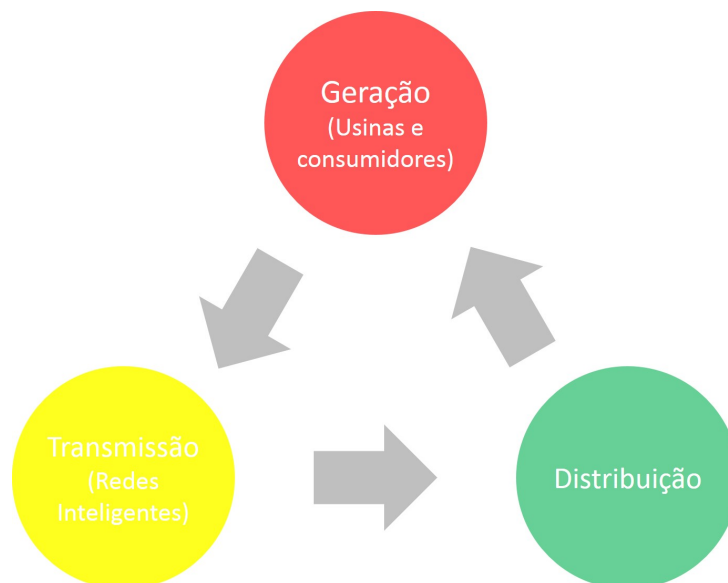


Figura 2.2 – Estrutura da rede inteligente.

2.1.1.2 Fluxo de dados

O fluxo de dados atual também é unidirecional, os medidores utilizados em grande parte das unidades consumidoras são eletromecânicos, apesar dessa realidade estar em fase de mudança, com a substituição para medidores eletrônicos, que, entretanto, permitem apenas a leitura do consumo de energia.

A arquitetura a ser desenvolvida com a implantação das redes inteligentes prevê um sistema operativo que possa atuar e se comunicar com todos os níveis do sistema elétrico, como pode ser observado na Figura 2.3. A operação e proteção confiável são características vitais para um funcionamento apropriado das redes inteligentes. Como “porta de entrada” dos consumidores estão os medidores, que quando inteligentes poderão fornecer informações a ambos os lados (gerador/consumidor), o que não ocorre na rede atual.

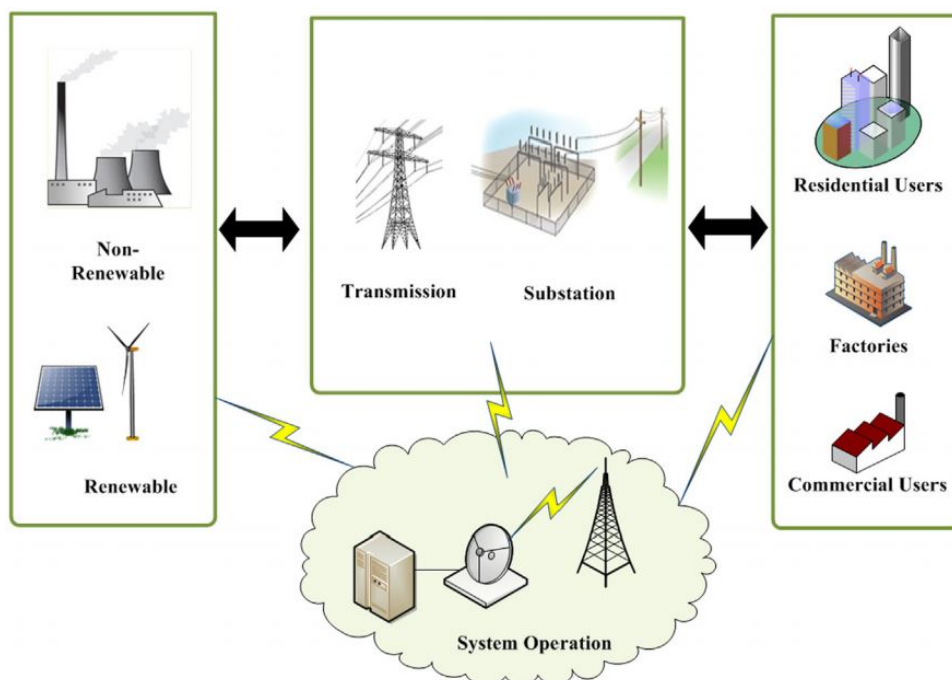
Os medidores inteligentes são um tipo de medidor avançado que mede o consumo de energia do cliente e fornece informações adicionais para a concessionária de energia, em comparação com um medidor de energia regular. Eles podem ler o consumo de energia (incluindo os valores de tensão, ângulo de fase e da frequência) e se comunicar de forma segura para transmissão de dados. Os medidores inteligentes permitem uma comunicação bidirecional de dados, dando a capacidade de recolher informações sobre o consumo e fornecimento de energia a partir de instalações de clientes. Um sistema de medição inteligente inclui um medidor inteligente, infra-estrutura de comunicação e dispositivos de controle (DEPURU et al., 2011).

2.1.1.3 Variedade de formas de geração

A Matriz Energética brasileira tem como base a geração a partir de usinas hidrelétricas (mostrada na Figura 2.4), pelo fato do país possuir uma grande quantidade de bacias hidrográficas aproveitáveis. Essa participação da geração por meio de hidreletricidade, segundo a ANEEL, correspondia a pouco mais de 60% em novembro de 2016, como mostrado na Figura 2.4, indicando uma participação menor que a registrada em 2013 quando correspondia a cerca de 70%.

Essa mudança de cenário dá-se por alguns fatores, como a diminuição dos custos de implantação dos parques eólicos, que eclodiram na última década, o tempo necessário para construção de uma usina hidrelétrica, como é o caso da usina de Belo Monte¹ que está em construção, assim como a entrada do consumidor fornecendo energia elétrica com a geração distribuída.

¹ A usina hidrelétrica de Belo Monte é um projeto do planejamento energético brasileiro, que está sendo implantado no rio Xingu, no estado do Pará, região Norte do Brasil. Quando finalizado, irá acrescentar pouco mais de 11 mil megawatts (MW) de capacidade instalada à matriz energética nacional (EMPRESA DE PESQUISA ENERGETICA, 2011).

Figura 2.3 – Arquitetura *Smart Grid*.

Fonte: [Fadaeenejad et al. \(2013\)](#)

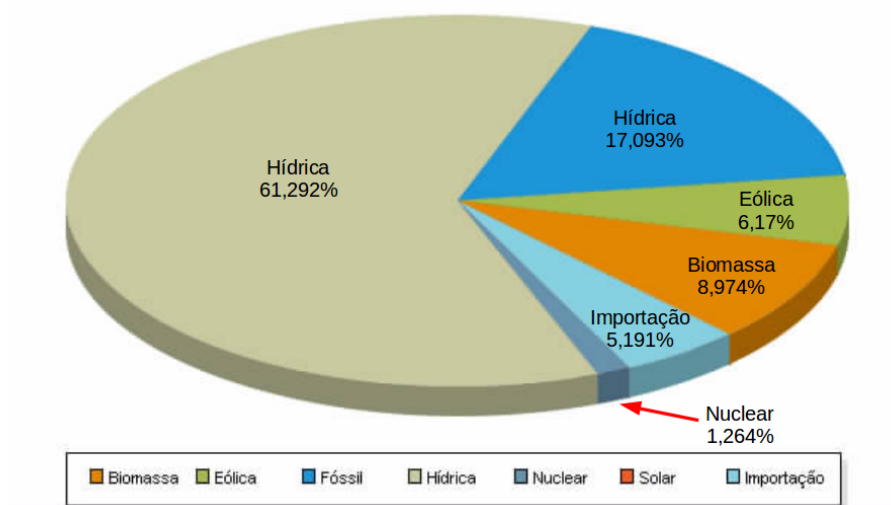


Figura 2.4 – Matriz Energética do Brasil – adaptada.

Fonte: [ANEEL \(2016\)](#).

O Brasil já tem como característica uma pluralidade no seu processo de geração de energia, inclusive com a adoção da resolução 482/2012 e posteriormente a 687/2015 da ANEEL, as mini e micro gerações distribuídas têm a possibilidade de tornar menor a dependência de formas de geração onerosas e com altas taxas de emissão de carbono, como é o caso das termelétricas, que hoje funcionam de forma complementar à matriz, isto é, quando os reservatórios de água das usinas hidrelétricas estão baixos elas operam

para suprir a demanda por energia.

2.2 NS-2

O NS-2 é um simulador de redes de computadores de código aberto, desenvolvido pela UC Berkeley, como parte do projeto VINT – *Virtual InterNetwork Testbed*. Sua primeira versão foi lançada em 1995 e um ano depois o NS-2 foi anunciado (PEREIRA, 2015). O software permite validar o desempenho de protocolos já existentes, simular redes, validar novos protocolos antes de serem implementados, simular sistemas de redes de simples a complexas, trabalhando também com a simulação de protocolos como o TCP e suas variantes, redes cabeadas, sem fio, roteamento de pacotes e satélites (ORTEGA, 2015).

Sua programação é realizada empregando duas linguagens, o C++ e OTcl – *Object-oriented Tool Command Language* (Linguagem de Comando de Ferramentas Orientada a objetos). A primeira tem como vantagem ser uma linguagem mais robusta, compilada, usada para a manipulação de *bytes*, pacotes e para a implementação de algoritmos, ou seja, o núcleo do programa. Já a segunda é uma linguagem de scripts, variação da linguagem TCL, desenvolvida pelo MIT – *Massachusetts Institute of Technology* (Instituto de Tecnologia de Massachusetts), orientada a objetos, utilizada como *frontend*. Por ser uma linguagem interpretada, não compilada, torna-se de mais fácil manipulação. A estrutura básica do NS-2 é mostrada na Figura 2.5.

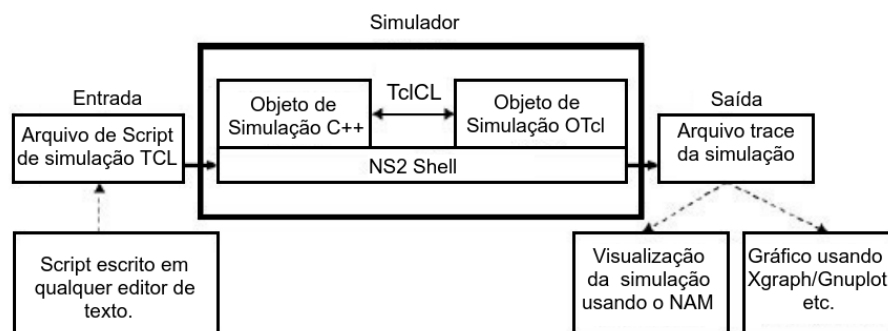


Figura 2.5 – Estrutura básica da simulação.

Fonte: Champati e Rajib (2016) adaptado.

2.2.1 A simulação no NS-2

A simulação inicia-se por *scripts* OTcl que, em seguida, sujeitam processos ao núcleo do NS-2 para execução (C++) e, por fim, geram os resultados da simulação que serão submetidos à análises (COUTINHO et al., 2003). O núcleo do NS-2, implementado em C++, utiliza-se das características da programação orientada a objetos, como classes,

herança, polimorfismo, etc. Sendo assim, o usuário poderá utilizar classes já existentes, de acordo com a característica da rede que se deseja simular, ou ainda realizar a implementação de novas classes.

Na descrição da classe, é criada uma ligação entre um objeto C++ do núcleo e um objeto OTcl, que pode ser vista em linhas de código na Figura 2.6.

```
static class ChannelClass : public TclClass {
public:
    ChannelClass() : TclClass("Channel") {}
    TclObject* create(int, const char*const*) {
        return (new Channel);
    }
} class_channel;
```

Figura 2.6 – Ligação OTcl – C++ da classe Channel.

A ligação pode também ser apresentada como diagrama de blocos, como mostrado na Figura 2.7, em que tem-se (CHAMPATI; RAJIB, 2016), para uso OTcl:

- Configurações de rede;
- Executar simulações com módulos NS2 existentes;

e, para uso C++:

- Modificar um ou mais módulos existentes;
- Criar novos módulos para implementar novos protocolos.

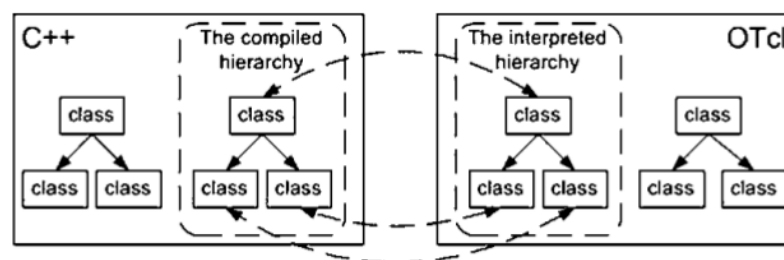


Figura 2.7 – Ligação OTcl – C++.

Fonte: Issariyakul e Hossain (2011)

A simulação utiliza a estrutura de camadas do modelo OSI, como mostrado na Figura 2.8, em que temos a presença de 5 nós interligados, agentes de transporte UDP – *User Datagram Protocol* (Protocolo de Datagrama de Usuário) e TCP – *Transmission Control Protocol* (Protocolo de Controle de Transmissão), com FTP – *File Transfer Protocol* (Protocolo de Transferência de Arquivos) enviando arquivos para o agente TCP e CBR –

Constant Bit Rate (Taxa de Bit Constante) para o agente UDP. Pode-se ver ainda a largura de banda dos enlaces (100Mbps, 54Mbps e 10Mbps) e o atraso para o envio, característico de cada enlace.

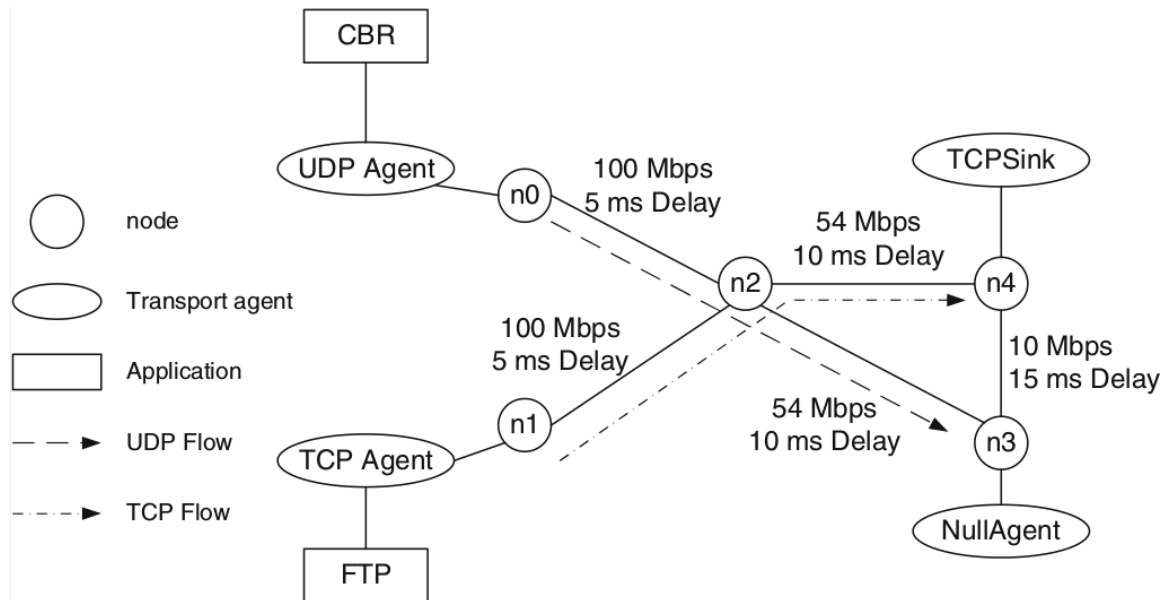


Figura 2.8 – Topologia simples de rede.

Fonte: [Issariyakul e Hossain \(2011\)](#)

2.2.2 Arquivo trace de saída

Os resultados da simulação são apresentados em um arquivo *trace* de saída, com a extensão *.tr*, que pode ser aberto em qualquer editor de textos comum, tendo uma estrutura como o mostrado na Figura 2.9, na qual:

- (A) Evento ocorrido, que pode ser uma entrada (+) ou saída (-) da fila, um descarte (d) ou recebimento (r) de pacote, etc;
- (B) Instante em que o evento ocorreu em segundos;
- (C) Nó de origem do evento;
- (D) Nó de destino do evento;
- (E) Tipo do pacote (TCP, UDP, ACK², DNP3, etc);
- (F) Tamanho do pacote em bytes;

² O ACK – *Acknowledgment* é o comando de reconhecimento, isto é a resposta de retorno numa comunicação TCP.

A	B	C	D	E	F	G	H	I	J	K	L
+	0.0001	0	3	tcp	40	-----	1	0.0	4.0	0	0
-	0.0001	0	3	tcp	40	-----	1	0.0	4.0	0	0
r	0.00142	0	3	tcp	40	-----	1	0.0	4.0	0	0
+	0.00142	3	4	tcp	40	-----	1	0.0	4.0	0	0
-	0.00142	3	4	tcp	40	-----	1	0.0	4.0	0	0
r	0.04174	3	4	tcp	40	-----	1	0.0	4.0	0	0
+	0.04174	4	3	ack	40	-----	1	4.0	0.0	0	1
-	0.04174	4	3	ack	40	-----	1	4.0	0.0	0	1
r	0.08206	4	3	ack	40	-----	1	4.0	0.0	0	1
+	0.08206	3	0	ack	40	-----	1	4.0	0.0	0	1
-	0.08206	3	0	ack	40	-----	1	4.0	0.0	0	1
r	0.08338	3	0	ack	40	-----	1	4.0	0.0	0	1
+	0.08338	0	3	tcp	1040	-----	1	0.0	4.0	1	2
-	0.08338	0	3	tcp	1040	-----	1	0.0	4.0	1	2
+	0.08338	0	3	tcp	1040	-----	1	0.0	4.0	2	3
-	0.0917	0	3	tcp	1040	-----	1	0.0	4.0	2	3
r	0.0927	0	3	tcp	1040	-----	1	0.0	4.0	1	2
+	0.0927	3	4	tcp	1040	-----	1	0.0	4.0	1	2
-	0.0927	3	4	tcp	1040	-----	1	0.0	4.0	1	2

Figura 2.9 – *Trace* de saída, resultado da simulação de rede empregando NS-2.

- (G) Flags utilizados para notificação de congestionamento, que são definidos no arquivo `trace.cc`;
- (H) Identificação do fluxo de pacotes;
- (I) Endereço da fonte do evento;
- (J) Endereço do destino do evento;
- (K) Número de sequência do pacote;
- (L) Número que identifica o pacote na rede.

O arquivo `trace` mostra o que ocorreu numa simulação. Por exemplo, no caso da perda de um pacote, o mesmo pode entrar na fila (+), sair (-) mas não ser recebido no destino (r), fazendo com que o pacote precise ser reenviado ao ser checado pelo protocolo TCP. Há também o caso do pacote ter sido descartado pelo congestionamento da fila, neste caso sera feita uma indicação com (d), indicando o descarte *drop*.

2.2.3 Objeto animado NAM

Os resultados de um arquivo `trace` podem ser utilizados pela ferramenta NAM – *Network Animator* do NS-2, que mostra de forma animada o que ocorre na comunicação, como o comportamento dos nós, fluxo, queda de link, fila, entre outras coisas.

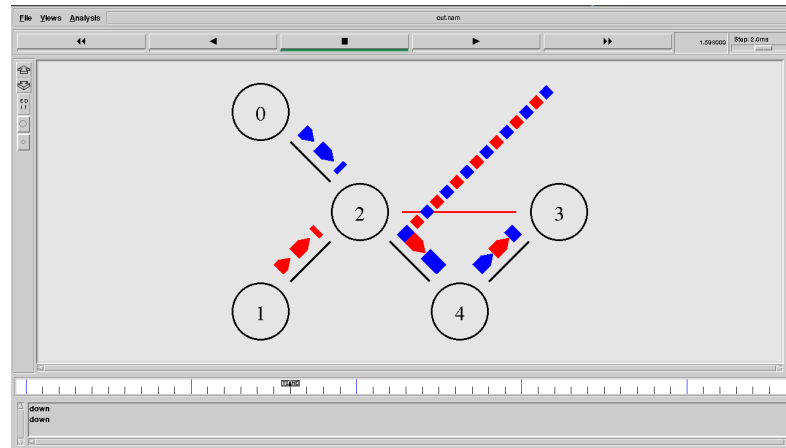


Figura 2.10 – Exemplo da animação do NS-2 – NAM.

Na Figura 2.10 tem-se um exemplo de utilização do objeto animado NAM, em que houve uma queda do enlace entre os nós 2 e 3 (indicado em vermelho), fazendo com que os pacotes utilizem um caminho alternativo (via nó 4), que por possuir uma largura de banda menor provoca um aumento da fila e, pela regra de tratamento da fila utilizado *Droptail*, faz com que alguns pacotes sejam descartados.

O grande interesse despertado pelo uso do *software* dá-se pela grande variedade do tipo de redes que ele pode simular, independente das já existentes, gerando assim novos objetos em C++ para o simulador, como a implementação feita por Pereira (2015) para redes ZigBee.

2.3 PLC

A tecnologia PLC, como o nome indica, é a transmissão de dados através de cabos de energia, sendo, na verdade, um termo genérico para qualquer tecnologia que usa a rede de energia elétrica como um canal de comunicação (LU et al., 2012). Segundo Ferreira e Lampe (2010), o primeiro sistema PLC era conhecido como RP – *Ripple Control* (Controle de Ondulação) e usava uma portadora de frequência entre 100 Hz e 1 kHz, compreendendo um sistema unidirecional com baixas taxas de dados, que tinha como desvantagem requerer vários Megawatts para transmissão de informação.

O primeiro uso dos sistemas PLC surgiram na década de 1920, sendo utilizados a partir dali pelas empresas de energia elétrica para telemetria³, controle remoto e comunicações de voz, por meio de equipamentos robustos e, normalmente, com uma vida útil superior a trinta anos (SOARES, 2010).

Uma das primeiras iniciativas para automação da rede elétrica foi feita usando a

³ Tecnologia focada em monitoramento, medição e/ou rastreamento por meio do envio de dados, via comunicação a uma central de controle.

tecnologia PLC (USMAN; SHAMI, 2013), porém os projetos de desenvolvimento dessas redes nunca envolveram requisitos de comunicação, de forma que a rede nunca foi projetada para transmissão de sinais de frequências diferentes do padrão de 50 ou 60 Hz. Além disso, ruídos excessivos gerados pelas diversas cargas elétricas, tais como a impedância variável da rede, contribuem para degradação do sinal PLC, fazendo da rede elétrica um meio hostil para transmissão de dados (COELHO, 2015).

2.3.1 Processo de modulação

O processo de modulação do sinal PLC é ilustrado na Figura 2.11, em que a informação (sinal de alta frequência) é adicionada a frequência da rede elétrica. Este sinal composto (energia elétrica + dados) é enviado pela rede elétrica e deve ser demodulado na recepção por meio de filtros e processadores de sinais, separando a energia elétrica do que é informação.

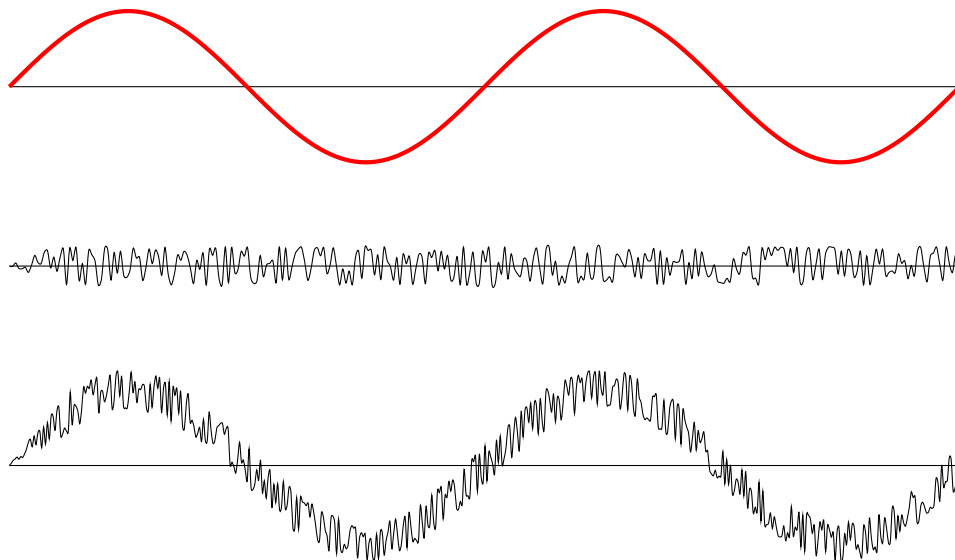


Figura 2.11 – Modulação da informação a ser transmitida.

Para realizar essa modulação, algumas técnicas são utilizadas, destacando-se:

- ***Spread Spectrum*** – Distribui a potência do sinal em uma ampla faixa;
- ***OFDM – Orthogonal Frequency Division Multiplex*** – Multiplexa o sinal de informação um número de bandas estreitas distribuídas lado a lado em frequências ortogonais;
- ***GMSK – Gaussian Minimum Shift Key*** – Mesma modulação da tecnologia GSM, um caso particular da modulação OFDM em que as portadoras são moduladas em fase, sendo robusto contra interferências de banda estreita;

- **FDM – Frequency-Division Multiplexing** – Usa modulação por onda contínua para colocar cada sinal em uma frequência específica da banda.
- **TDM – Time-Division Multiplexing** – Multiplexa o sinal em diferentes janelas de tempo.
- **CDM – Code-Division Multiplexing** – Diferentes sinais são transmitidos simultaneamente na mesma banda de frequências, separados por correlação de códigos.

2.3.2 Frequência de operação

A diferença em relação ao espectro fundamental divide o PLC em duas classes principais (SATO et al., 2015), (GALLI et al., 2010):

- Banda larga (*BroadBand – BB*), que opera em frequências altas, em torno de 2 a 100MHz, utilizada para comunicação de dados e voz, VoIP⁴, jogos, internet de alta velocidade, etc;
- Banda estreita (*NarrowBand – NB*), que opera normalmente em frequências baixas, de 3 a 500kHz, aplicado em redes inteligentes. Vários padrões PLC e tecnologias foram desenvolvidos nesses dois grupos, como pode ser observado na Figura 2.12.

2.3.2.1 Padrões, Especificações e Regulamentação de Frequência

Com o desenvolvimento e disseminação das tecnologias PLC, os órgãos regulamentadores dos países e regiões adotaram regras para padronizar os espectros de frequências utilizados, bem como do nível de sinal injetado na rede, resumidas na Tabela 2.1:

Na Europa, a CENELEC apresenta um desenvolvimento mais maduro, dividindo o espectro de frequências em quatro bandas: A (3 – 95kHz), B (95 – 125kHz), C (125 – 140kHz) e D (140 – 148,5kHz). A primeira é utilizada para fornecedores de eletricidade e seus associados, a B e C são limitadas ao consumidor, sendo necessário o uso de protocolos de acesso ao meio para utilização do espectro C, já o D é livre para o consumidor (COELHO, 2015).

No Brasil, para o uso de banda larga, tem-se a resolução 527 da Anatel, de 8 de abril de 2009, que regulamenta o uso da rede elétrica para aplicações PLC, utilizando as faixas de frequência de 1,7 a 50MHz, existindo as faixas de exclusão para os espectros de frequências atribuídos para o Serviço Móvel Aeronáutico e Radioamador, mostrado na Tabela 2.2, sendo também proibido o uso na faixa de frequências de 2,1735–2,1905MHz dentro de zonas de proteção de estações costeiras.

⁴ *Voice over Internet Protocol*, é a transmissão da voz humana utilizando a rede de dados baseada em Protocolo de Internet (GOODE, 2002).

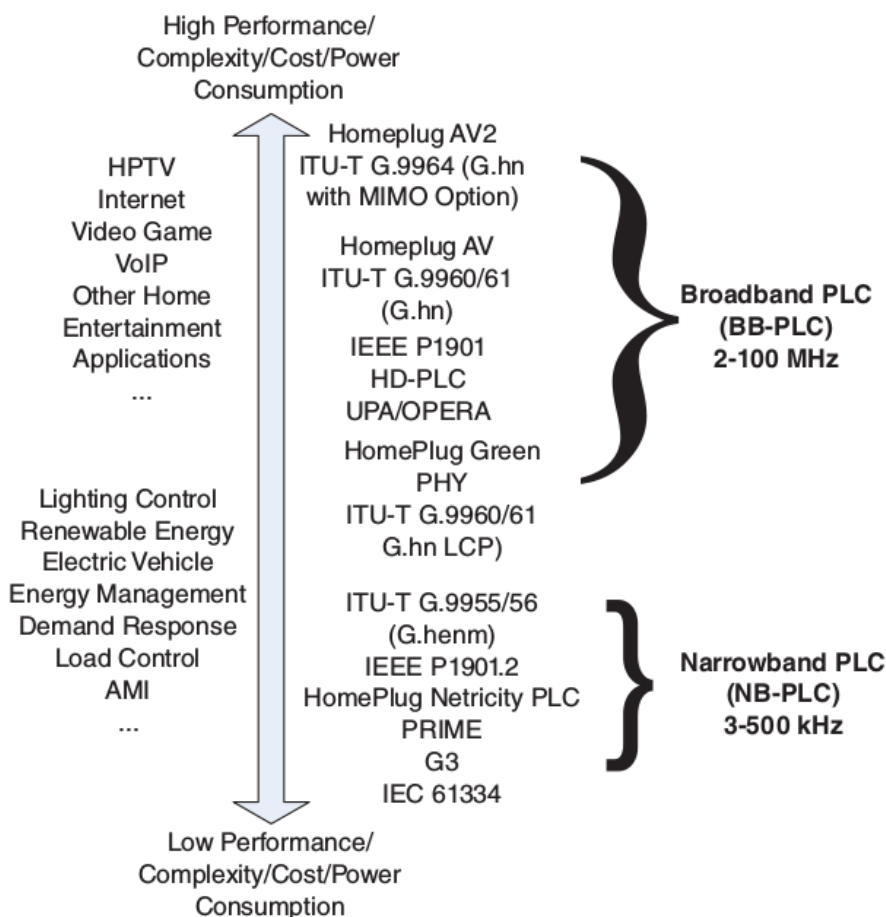


Figura 2.12 – Vários padrões de tecnologia PLC nos grupos NB-PLC e BB-PLC.

Fonte: Sato et al. (2015).

Ainda referente às zonas de proteção de estações costeiras, nas faixas de radio-frequências listadas na Tabela 2.3, os limites de radiação indesejada, causada pelos sistemas PLC nas redes de média tensão, devem ser atenuados a um nível de, no mínimo, 10dB abaixo dos limites das outras áreas.

Apesar de haver regulamentação por parte da ANATEL quanto ao uso de banda larga utilizando o PLC nas redes elétricas, segundo [Coelho \(2015\)](#), no Brasil ainda não há qualquer regulamentação quanto ao uso de banda estreita abaixo de 150kHz, mas com a estimativa de que o país siga a tendência de uso do padrão europeu, tendo em vista que boa parte de sua regulamentação segue padrões internacionais.

2.3.3 PLC em Redes Inteligentes

Na integração com redes inteligentes, o PLC pode ser utilizado em diversas atividades, como na leitura automática de medidores, supervisão e gerência da demanda, análise de sobrecargas, notificação de quedas, supervisão de perda de fase, caracterização de falhas, tele-controle de subestações de transformadores, leitura remota de parâmetros de energia

Tabela 2.1 – Regulação da Compatibilidade Eletromagnética – EMC para BB e NB-PLC.

EMC regulations related to NB-PLC		
Countries/regions	Frequency bands	Responsible institutions/SDOs
United States	10–490 kHz	Federal Communications Commission (FCC)
Japan	10–450 kHz	Association of Radio Industries and Business (ARIB)
European Union	3–148.5 kHz	European Committee for Electrotechnical Standardization (CENELEC)
China	3–90 kHz	China Electric Power Research Institute (CEPRI)
EMC regulations related to BB-PLC		
EU	1.6–30 MHz	CENELEC (CENELEC EN EN 50561-1)
Worldwide	Limits specified for 9 kHz to 400 GHz [10]	International Electrotechnical Commission (CISPR 22 ed6.0)

Fonte: [Sato et al. \(2015\)](#).

Tabela 2.2 – Faixas de exclusão para uso do PLC no Brasil.

Faixas de Frequências (MHz)
2,754–3,025
3,400–3,500
4,453–4,700
5,420–5,680
6,525–6,876
6,991–7,300
8,815–8,965
10,005–10,123
11,275–11,400
13,260–13,360
13,927–14,443
17,900–17,970
21,000–21,450
21,924–22,000
28,000–29,700

Fonte: [ANATEL \(2009\)](#).

elétrica (tensão, corrente, potência, etc...), podendo inclusive ser compartilhados, caso conveniente, com a concessionária de distribuição de água e gás ([FRANÇA et al., 2006](#)).

Além desses, outros parâmetros possíveis descritos por [França et al. \(2006\)](#) são:

- Autorização e desconexão remotas de interruptores e dispositivos pelos usuários (razões de segurança) e controle operacional de ações remotas;

Tabela 2.3 – Faixas de Radiofrequência relativas à zona de proteção de estações costeiras.

Faixas de Frequências (MHz)
4,122–4,128
4,177–4,178
4,207–4,208
6,212–6,218
6,268–6,269
6,312–6,313
8,288–8,294
8,364–8,365
8,376–8,377
12,287–12,293
12,520–12,521
12,577–12,578
16,417–16,423
16,695–16,696
19,680–19,681
22,376–22,377
26,100–26,101

Fonte: [ANATEL \(2009\)](#).

- Mudança remota de parâmetros contratuais sem necessidade de acesso ao medidor (por exemplo, potência máxima);
- Comparação entre a energia fornecida pelos transformadores e a energia consumida, identificando mais rapidamente a presença dos desvios de energia, conhecidos como “gatos”;
- Detecção de roubos e prevenção de uso não autorizado de medidores;
- Supervisão da qualidade de serviço fornecida a cada consumidor (por exemplo, número e extensão das interrupções de serviço);
- Previsão de consumo, gerência de consumo, e serviço automatizado dos usuários;
- Possibilidade de aplicação da tarifa branca de energia e da geração distribuída;
- Erros de leitura e diminuição das disputas legais com os usuários;
- Desconexão imediata, desde que autorizada pela regulamentação.

A tecnologia PLC fornece uma plataforma de comunicação bidirecional capaz de fornecer dados em tempo real para uma variedade de aplicações de serviços públicos, permitindo assim identificar e até mesmo prever falhas de equipamento e sua localização, tendo como características o uso dos protocolos IP, com alta da disponibilidade de largura

de banda, tanto larga como estreita. Pela experiência e maturidade, o uso dessa tecnologia apresenta uma maior aptidão na distribuição da rede, setor de maiores transformações na transição para uma rede inteligente, também por atuar ativamente com as microrredes, geração distribuída (GD), e participação dos consumidores (GALLI et al., 2010).

2.3.3.1 PLC em redes de média tensão

As distribuidoras de energia elétrica utilizam a média tensão⁵ para levar ao consumidor a energia proveniente das linhas de transmissão. Essa etapa do processo de fornecimento de energia necessita de um grande volume de transmissão de dados referente ao estado do equipamento e condições de fluxo de potência que devem ser transferidos entre subestações dentro da rede. Essas subestações num passado recente, não eram equipadas com capacidades de comunicação, o que tem mudado com a implantação de novas subestações, algumas usando a tecnologia PLC.

Na automação de subestações algumas de suas funções precisam de IEDs internos para a comunicação com IEDs externos, como no caso da localização do defeito, isolamento de falhas e restauração do serviço. Dentre os IEDs externos estão interruptores, religadores e seccionadoras. Em outro exemplo, a liberação de tensão no sistema de distribuição requer comunicações entre IEDs da subestação e IEDs de alimentadores de distribuição atendidos pela subestação. Todas estas comunicações exigem conectividade de baixa velocidade que está bem dentro das capacidades do NB-PLC (GALLI et al., 2010).

Essa comunicação entre subestações utiliza comumente o protocolo baseado na norma IEC 61850 – *International Electrotechnical Commission* (Comissão Eletrotécnica Internacional), que prevê a modelagem dos dispositivos de automação da subestação, ou ainda utiliza o protocolo DNP3. A norma IEC 61850 se desenvolve na camada de aplicação do modelo OSI, considerada muito mais que um protocolo de comunicações, ou seja, uma arquitetura de automação de subestações.

Ainda dentro do sistema de distribuição, tem-se o sistema *outdoor* (fora de casa), que acessa o *Backbone*⁶, fazendo, dentre outras coisas, que a rede elétrica consiga ter acesso à Internet, sendo composto pela linha que vai desde o transformador de distribuição até o medidor de energia, com uma conexão realizada por meio de um equipamento denominado **mestre** (SILVA, 2009).

Esse equipamento faz a interface entre a rede elétrica de distribuição e o sistema de telecomunicações, transformando o sinal do equipamento de telecomunicações em sinal modulado, injetando sobre a rede elétrica, normalmente instalado junto ao transformador, tendo em alguns casos a necessidade de instalação de repetidores devido à distância. Esses

⁵ Segundo a ANSI/IEEE 1585-2002, a média tensão é definida como a faixa de tensão em corrente alternada de 1kV a 35kV.

⁶ Espinha dorsal que interconecta os dispositivos numa rede (FOROUZAN, 2009).

repetidores recuperam e reinjetam o sinal transmitido (SILVA, 2009).

Rossi (2013) diz ainda que essa comunicação utiliza tipicamente frequências portadoras em paralelo para distribuir os dados em uma faixa de frequência maior, permitindo que portadoras individuais estejam ligadas / desligadas com um menor impacto em toda a rede de dados. Isto faz a comunicação ser menos vulnerável a erros ou interferência de ruídos.

2.3.3.2 PLC em redes de baixa tensão

Galli et al. (2010) diz que, quando dispositivos PLC não interoperáveis compartilham os mesmos cabos de rede utilizando as mesmas frequências ao mesmo tempo, a interferência mútua gerada provoca uma degradação grave do desempenho ou interrupção do serviço e, como já citado, pelo fato de haver o compartilhamento dos cabos com a transmissão de dados e energia, os sinais que são gerados no interior das instalações podem interferir com os sinais gerados fora delas, mesmo existindo mecanismos de coexistência que limitam a interferências prejudiciais causadas pelos dispositivos vizinhos. Um exemplo da aplicação da tecnologia PLC pode ser visto na Figura 2.13.

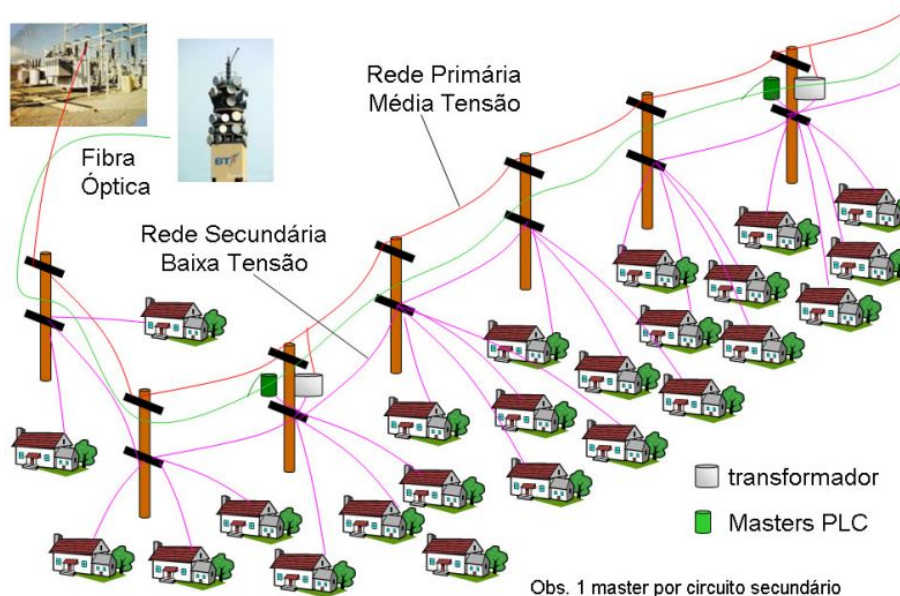


Figura 2.13 – Tecnologia PLC para baixa tensão.

Fonte: Rossi (2013).

Ainda na rede de baixa tensão, porém dentro das edificações (residenciais, comerciais e industriais), é utilizado um sistema denominado *indoor* (dentro de casa), abrangendo o trecho que vai desde o medidor de energia do usuário até todas as tomadas no interior do estabelecimento, sendo constituída pela rede de distribuição elétrica nas instalações já existentes e pelos modems para conexão dos equipamentos que serão interligados ao

serviço. Esse uso possibilita ampliar gradativamente o espectro das redes prediais, pois transforma todas as tomadas em pontos de acesso de sinais de dados para computadores pessoais, telefones e impressoras, bem como para outros dispositivos eletroeletrônicos com esse tipo de facilidade (SILVA, 2009).

2.4 DNP3

O protocolo DNP3, ou IEEE 1815, é um protocolo orientado a eventos que segundo Ramalho et al. (2013), foi desenvolvido para ser o primeiro protocolo aberto fornecido as concessionárias de serviços públicos. Por ter sido desenvolvido por empresas de integração de sistemas, os projetistas do DNP3 tentaram incorporar os melhores recursos de todos os protocolos de utilitário que encontraram (IEEE, 2012).

2.4.1 Recursos do protocolo

Dentre os vários recursos que podem ser utilizados pelo protocolo, destacam-se (IEEE, 2012):

- **Radiodifusão** – Capacidade de enviar uma única mensagem para vários dispositivos;
- **Selecionar antes de operar** – Capacidade de escolher usar ou não confiabilidade extra ao operar uma saída;
- **Dados com precisão de tempo** – Alguns dos protocolos de utilidade mais populares, como Modbus, não possuem dados com precisão de tempo, levando ao desenvolvimento de soluções proprietárias quando da necessidade deste recurso. Outros protocolos suportam dados com precisão de tempo somente em dados binários. DNP3 permite este recurso em quase todos os dados, tornando-se cada vez mais útil em aplicações em que dados são reunidos de forma progressiva, como log de “sequência de eventos”;
- **Sincronização precisa do tempo** – Muitos protocolos anteriores não tinham como contabilizar atrasos de transmissão e de software durante a sincronização. O método utilizado no DNP3 é uma fusão de várias soluções de protocolos diferentes;
- **Flags de qualidade** – Mecanismo para ver se os dados eram válidos e por quê. Alguns protocolos, projetados por fornecedores de IEDs cujos dados estavam sempre online, não incluíam esse recurso;
- **Múltiplos formatos de dados** – Capacidade de relatar dados em uma variedade de formatos: 16 bits, 32 bits, com *flag*, sem *flag*, ponto flutuante, decimal codificado em binário (BCD), compactado, descompactado e assim por diante;

- **Grupos de busca** – Capacidade de definir e solicitar um conjunto grande de dados não relacionados usando um único pedido;
- **Separação de camadas** – Separação da função de “obter os dados lá” a partir das funções reais SCADA;
- **Relatório por exceção** – Mais do que qualquer outro recurso, a capacidade de relatar apenas as mudanças nos dados ajudou a tornar o DNP3 bem-sucedido;
- **Indicações internas** – Como vários protocolos que são mais recentes do que DNP3 descobriram, é extremamente útil ter um conjunto global de sinalizadores retornados em cada resposta. Esses sinalizadores indicam a integridade do dispositivo e os resultados da última solicitação.

2.4.2 Estrutura do protocolo

O protocolo foi desenvolvido seguindo o modelo EPA – *Enhanced Performance Architecture* (Arquitetura de desempenho aprimorada), que tem um número reduzido de camadas, desenvolvido para operar com apenas três das sete camadas do modelo de referência OSI, sendo elas a camada física, enlace e aplicação (IEEE, 2012), como pode ser observado na Figura 2.14.

Camada		
Aplicação	1	Aplicação
Apresentação	2	
Seção	3	
Transporte	4	
Rede	5	
Enlace	6	Enlace
Física	7	Física
Modelo OSI		Modelo EPA

Figura 2.14 – Comparação entre o modelo de referência OSI e o modelo EPA.

Da Figura 2.15 pode-se verificar também que os dispositivos apresentam uma separação de camada de usuário, camada de aplicação, função de transporte e enlace de dados, que possuem as seguintes funcionalidades apresentadas a seguir, complementadas pela Tabela 2.4 (IEEE, 2012):

- a) **Camada do usuário** – Não indica exatamente uma camada, mas sim um software do usuário DNP3, que é o programa de aplicação que torna um dispositivo único, seja um mestre, IED ou um concentrador de dados. Ele faz uso dos serviços da camada de aplicação para enviar/receber mensagens de/para outro dispositivo DNP3.
- b) **Camada de Aplicação** – No dispositivo mestre, essa camada é responsável por receber as requisições efetuadas pelo *software*, sendo responsável pela interface entre a rede DNP3 e o usuário.
- c) **Função de transporte** – Executa uma função semelhante a uma camada de particionamento de mensagens grandes em mensagens menores que a camada de enlace de dados é capaz de manipular. A função de transporte é por vezes referida como uma “pseudocamada”.
- d) **Camada de enlace** – Tem como principal função verificar possíveis erros e retornar confirmação de entrega de datagramas ou ACK *Acknowledgement* (confirmação).

Tabela 2.4 – Características das camadas DNP3.

Camada ou Função	Nome da unidade	Informações
Camada de Aplicação	Fragmento de aplicação	Permite a definição de um limite superior para os requisitos de memória para a recepção de mensagens. As solicitações devem caber em um único fragmento. As respostas podem exigir mais de um fragmento.
Função de Transporte	Segmento de transporte	A segmentação quebra um fragmento de enlace de dados em partes que se encaixam em uma moldura de enlace de dados. Cada segmento contém um cabeçalho de transporte, mas apenas o primeiro segmento de qualquer fragmento contém um cabeçalho de aplicação. Cada segmento pode ter um máximo de 250 octetos, incluindo o cabeçalho de transporte.
Camada de enlace de dados	Quadro de enlace	Um quadro pode ter até 292 octetos, incluindo seu cabeçalho e octetos CRC. Os quadros são projetados para a detecção de erros superior.

Fonte: [IEEE \(2012\)](#) – Adaptada.

2.4.3 Dispositivos

O DNP3 trabalha com dois tipos de dispositivos, mestre e estação remota *outstation* apresentados na Figura 2.15. Essa configuração do tipo Mestre – Escravo opera da seguinte forma:

- a) Mestre – Verifica periodicamente os outros dispositivos, por meio de verificação cíclica, fazendo o escoamento dos dados para o administrador;
- b) Estação remota – Faz a coleta, armazenagem e envio dos dados quando solicitado pelo dispositivo mestre ou quando identifica um evento crítico que deve ser reportado, chamado de mensagem não solicitada.

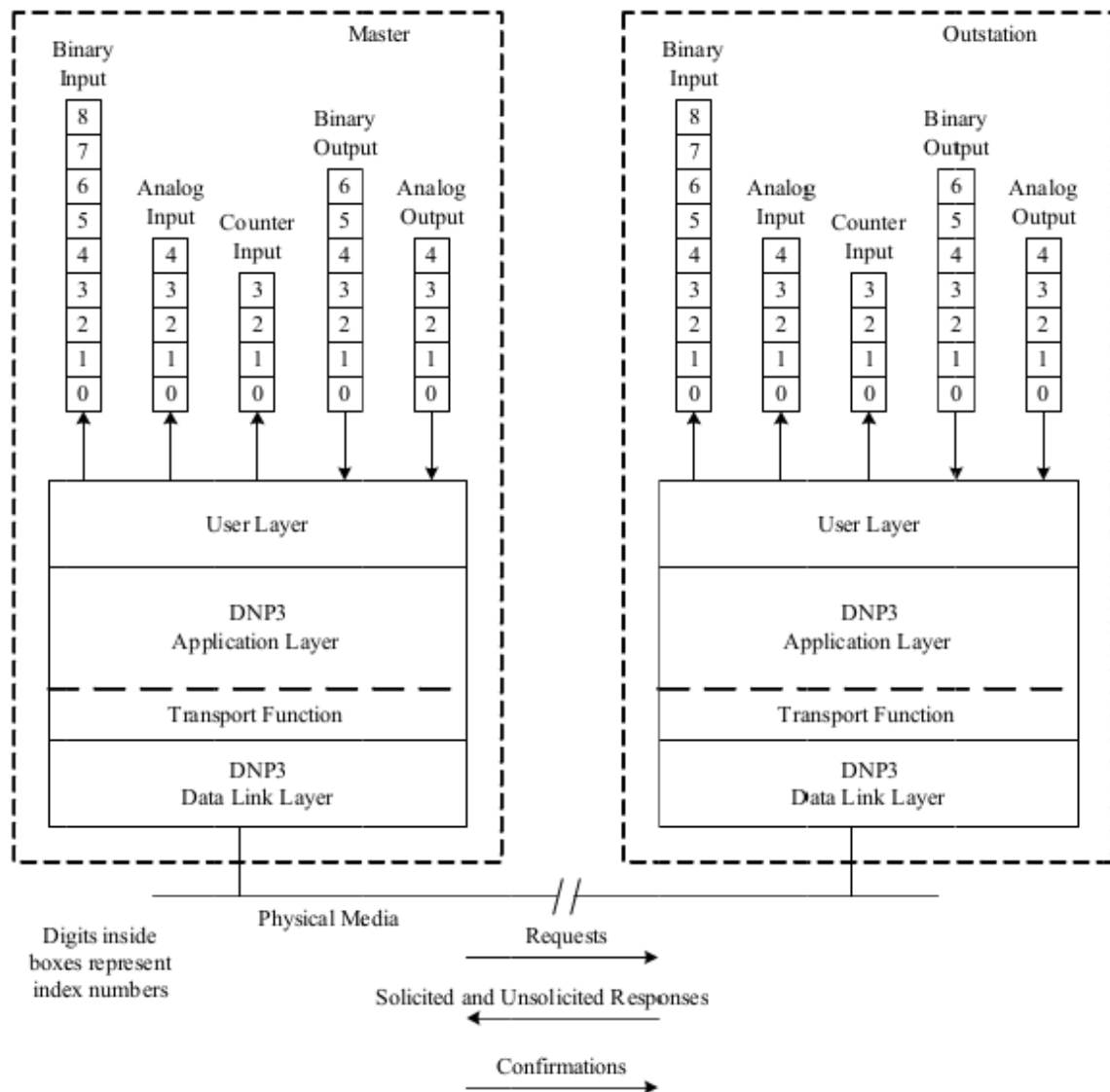


Figura 2.15 – Modelo mestre – escravo do protocolo DNP3.

Fonte: [IEEE \(2012\)](#).

O modelo de transmissão de dados entre as estações é apresentado na Figura 2.16.

2.4.4 Construção e troca de mensagens

As mensagens seguem a estrutura mostrada na Figura 2.17, em que a mensagem é passada desde a camada mais alta (camada de aplicação) na qual é fragmentada, passa pela

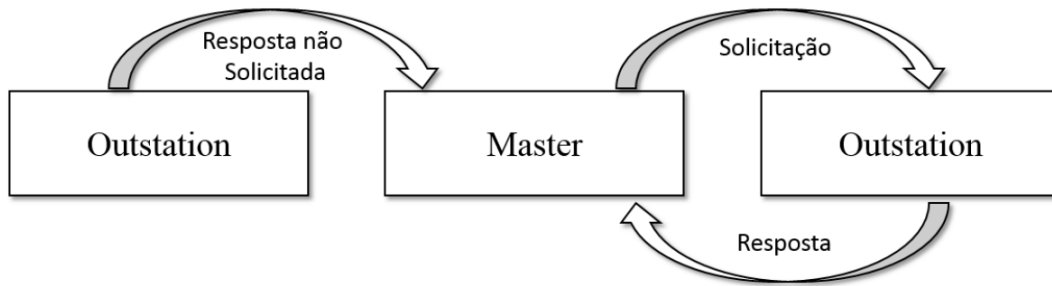


Figura 2.16 – Comunicação entre dispositivos.

Fonte: [Pereira \(2015\)](#).

função de transporte na qual é segmentada e, por fim, na camada inferior é dividida em quadros, agregando em cada uma das camadas a estrutura de cabeçalho, tornando-a maior a medida que avança por cada camada. Essa estrutura é explicada por [Forouzan \(2009\)](#) fazendo uma analogia com a troca de cartas entre dois amigos, em que são adicionados o envelope, selo, carimbo, como se fossem as estruturas de cabeçalho da mensagem, cujo conteúdo importante é o que está inserido dentro dela.

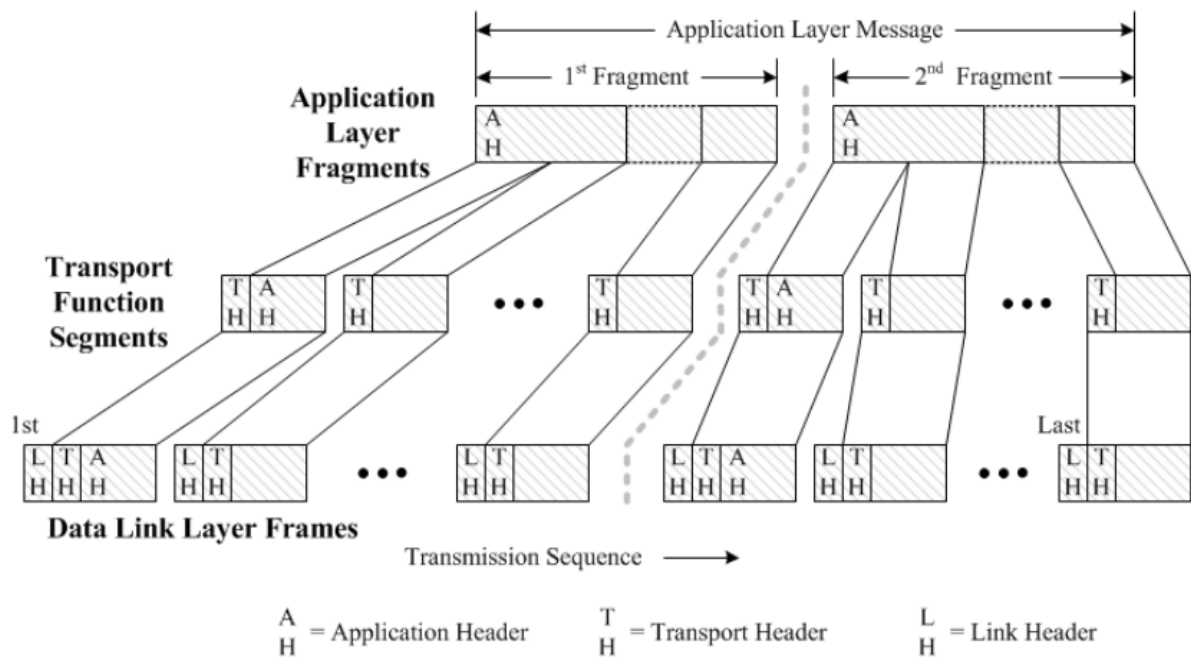


Figura 2.17 – Fragmentos, segmentos e quadros da mensagem DNP3.

Fonte: [IEEE \(2012\)](#).

Na camada de aplicação não há limite para o tamanho da mensagem, formando um bloco de dados administráveis chamado ASDU – *Application Service Data Unit* (Unidade de Dados de Serviço de Aplicação) que pode ser de qualquer tamanho, inclusive 0. Esses dados são fragmentados ainda nessa camada para serem encaminhados para a função de

transporte com tamanho máximo de 2048 bytes, sendo então denominadas de APDUs – *Application Protocol Data Unit* (Unidade de Dados de Protocolo de Aplicação), aumentando o tamanho do cabeçalho de dados denominado de APCI – *Application Protocol Control Information* (Controle de Informação de Protocolo de Aplicação), de 2 bytes para o caso de solicitação de mensagem a 4 bytes para resposta (ORTEGA, 2015).

Ao chegar à camada pseudo transporte, os fragmentos APDUs são encapsulados como unidades de serviços de transporte, nessa camada denominados de TSDU – *Transport Service Data Unit* (Unidade de Dados de Serviço de Transporte), que é segmentado em unidades menores denominadas TPDUs – *Transport Protocol Data Units* (Unidade de Dados de Protocolo de Transporte), com tamanho máximo de 249 bytes de dados e 1 byte de cabeçalho, totalizando um tamanho máximo de 250 bytes que será encaminhado para a camada de enlace de dados.

Na camada de enlace, os segmentos TPDUs são encapsulados como unidade de serviços de enlace, denominados de LPDUs – *Link Protocol Data Unit* (Unidade de Dados de Protocolo de Enlace), nos quais são adicionados 10 bytes de cabeçalho e 32 bytes de correção de erros, o CRC – *Cyclic Redundancy Check* (Verificação Cíclica de Redundância), totalizando 292 bytes a ser encaminhado pelo meio físico. Essa estrutura com indicação dos tamanhos é apresentada na Figura 2.18 e ilustrada por Jaimes (2012) na Figura 2.19 com um resumo dos tamanhos dos pacotes nas diferentes etapas.

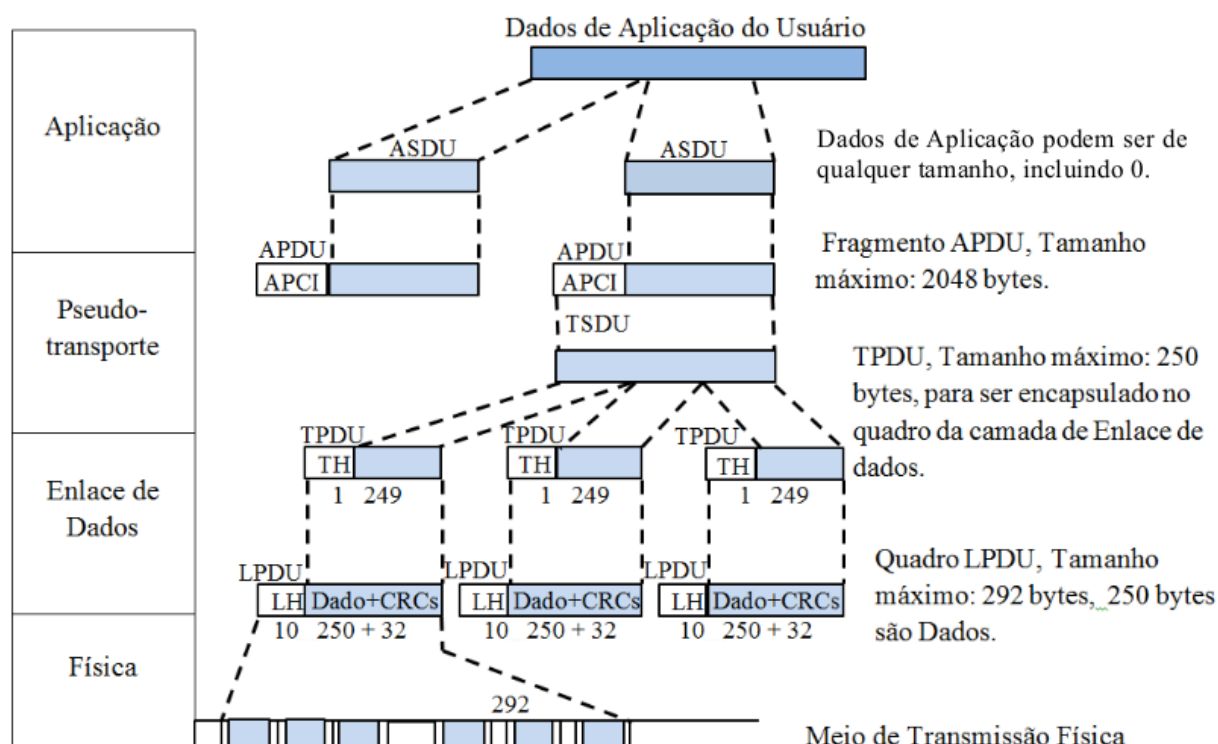


Figura 2.18 – Estrutura de construção de mensagem DNP3.

Fonte: Ortega (2015).

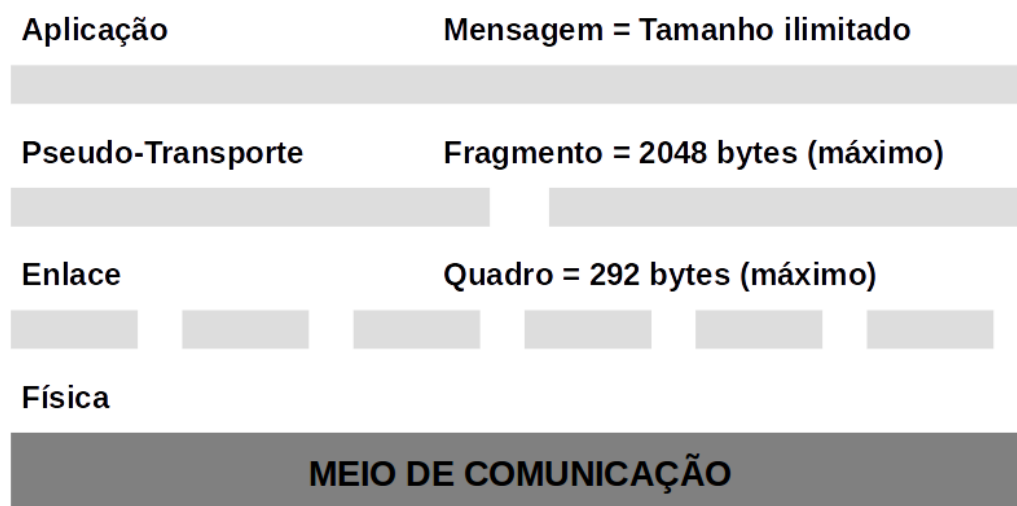


Figura 2.19 – Pacotes nas camadas do protocolo DNP3.

Fonte: [Jaimes \(2012\)](#) – Adaptado.

A troca de mensagens entre dispositivos é apresentada na Figura 2.20 na qual o mestre envia um comando para o escravo e são realizadas trocas de mensagens ao longo do tempo (eixo vertical) com confirmações de recebimento entre as estações.

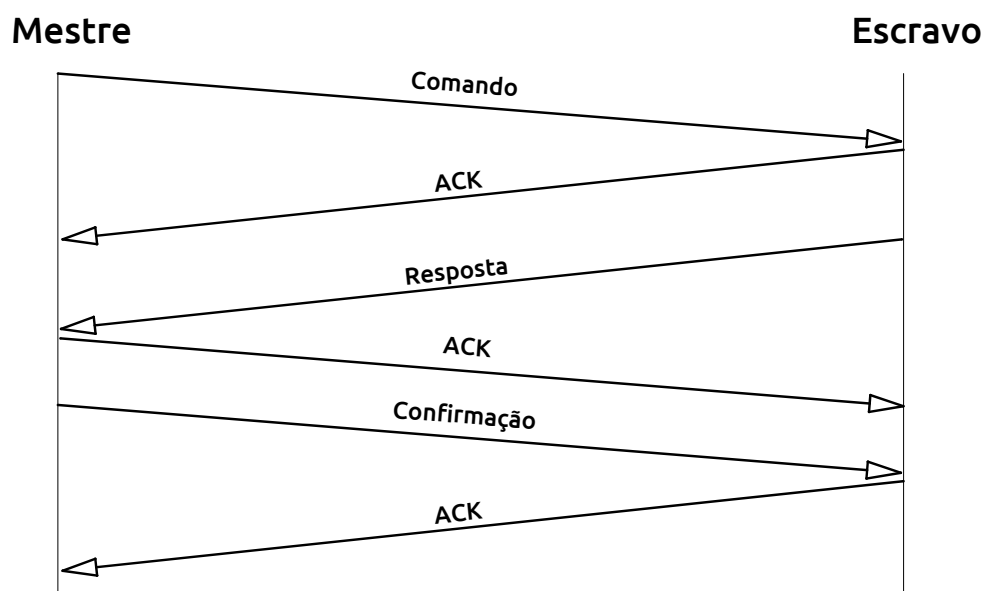


Figura 2.20 – Troca de datagramas no envio de um comando.

2.4.5 Topologias

O protocolo DNP3 apresenta quatro topologias distintas (CLARKE; REYNOLDS, 2004), que podem ser vistas na Figura 2.21 e descritas a seguir:

- i. **Ponto a ponto** – É a arquitetura mais simples, em que o dispositivo mestre tem uma conexão direta com o escravo;
- ii. **Multiponto** – A estação mestre se comunica e gerencia várias estações remotas, realizando uma comunicação ponto a ponto com cada um deles, sendo que a estação central solicita dados sequenciais a cada uma das estações remotas (ORTEGA, 2015);
- iii. **Hierárquico** – Funciona com um concentrador de dados intermediário, isto é, um dispositivo pode funcionar como mestre para uma sequência de escravos e escravo para outro dispositivo com nível hierárquico superior;
- iv. **Múltiplos mestres** – Neste caso têm-se vários dispositivos mestres distribuídos ao longo da linha, permitindo que operadores regionais obtenham informações de forma mais rápida, quando necessário.

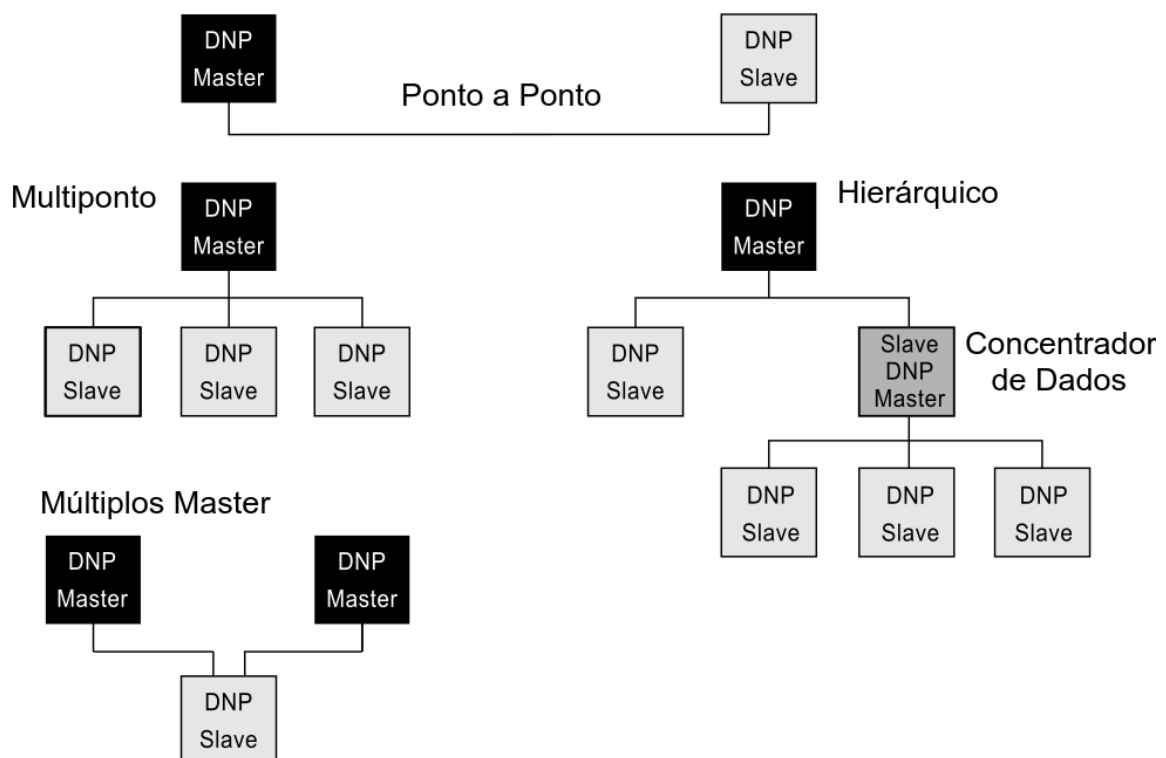


Figura 2.21 – Topologias do protocolo DNP3.

Fonte: Clarke e Reynolds (2004) – Adaptado.

2.4.6 Integração com o TCP/IP

Como o DNP3 foi originalmente concebido, sua camada física foi especificada como usando comunicação serial RS-232C, o que era uma escolha natural para a época, como na utilização de um ambiente de subestação com um grande número de dispositivos, como medidores, relés e outros IEDs.

Com o desenvolvimento dos sistemas de comunicação, houve uma pressão para a utilização do DNP3 sobre um ambiente de rede, e isso resultou na extensão da especificação DNP3 para incluir isso. O Comitê Técnico do Grupo de Usuários DNP3 definiu um método para transportar o protocolo envolvendo o uso do conjunto de protocolos de Internet para as camadas de transporte e rede e a camada física Ethernet.

A ideia de transportar DNP3 em um ambiente de rede envolve o encapsulamento dos quadros de dados da camada de ligação de dados DNP3 dentro dos quadros de camada de transporte do conjunto de protocolos de Internet e permitindo que a pilha de protocolos forneça os quadros de camada de ligação de dados DNP3 ao local de destino no local da camada física DNP3 original. O método seguinte foi recomendado pelo Comitê Técnico (IEEE, 2012):

- a) DNP3 deve usar o pacote IP para transportar mensagens de LAN / WAN;
- b) O enlace físico recomendado é Ethernet (porém outros podem ser usados);
- c) Todos os dispositivos devem suportar TCP e UDP;
- d) TCP deve ser usado para WANs;
- e) É altamente recomendado o uso de TCP para LANs;
- f) UDP pode ser usado para LANs altamente confiáveis;
- g) Para o envio de mensagens de difusão, é preciso utilizar a comunicação UDP;
- h) A pilha de protocolos DNP3 deve ser mantida na íntegra.

Para obter a interoperabilidade entre versões de diferentes fabricantes, é necessário garantir que as implementações de todos os fabricantes suportem os mesmos objetos e funções de dados. No entanto, exigir que todos os fabricantes suportem todos os objetos de dados e funções não seria muito eficiente. Nesse sentido foram definidos três níveis de subconjuntos no DNP3. Estes são designados no formato “*DNP3 Application Layer Protocol Level x*” (Nível de Protocolo da Camada de Aplicação), ou simplesmente abreviado como DNP3-L1, L2 ou L3. Os níveis são descritos na Tabela 2.5 (CLARKE; REYNOLDS, 2004).

Tabela 2.5 – Níveis de implementação do protocolo DNP3.

Nível	Descrição
Nível 1	É destinado ao uso entre uma estação principal ou um concentrador de dados, e um dispositivo terminal IED pequeno, que pode ser um relé, um medidor ou um controlador independente de qualquer tipo.
Nível 2	Define um subconjunto maior de recursos DNP3 do que o Nível 1. Destina-se a ser usado entre uma estação mestre ou um concentrador de dados e uma RTU <i>Remote Terminal Unit</i> (Unidade Terminal Remoto) ou um IED grande.
Nível 3	Define o maior subconjunto de recursos DNP3. Ele não requer suporte de todos os recursos possíveis do DNP3, mas cobre a maioria dos recursos mais frequentemente necessários. Isto é implementado tipicamente entre uma estação mestre e uma RTU maior ou mais avançada.

Fonte: [Clarke e Reynolds \(2004\)](#) – Adaptada.

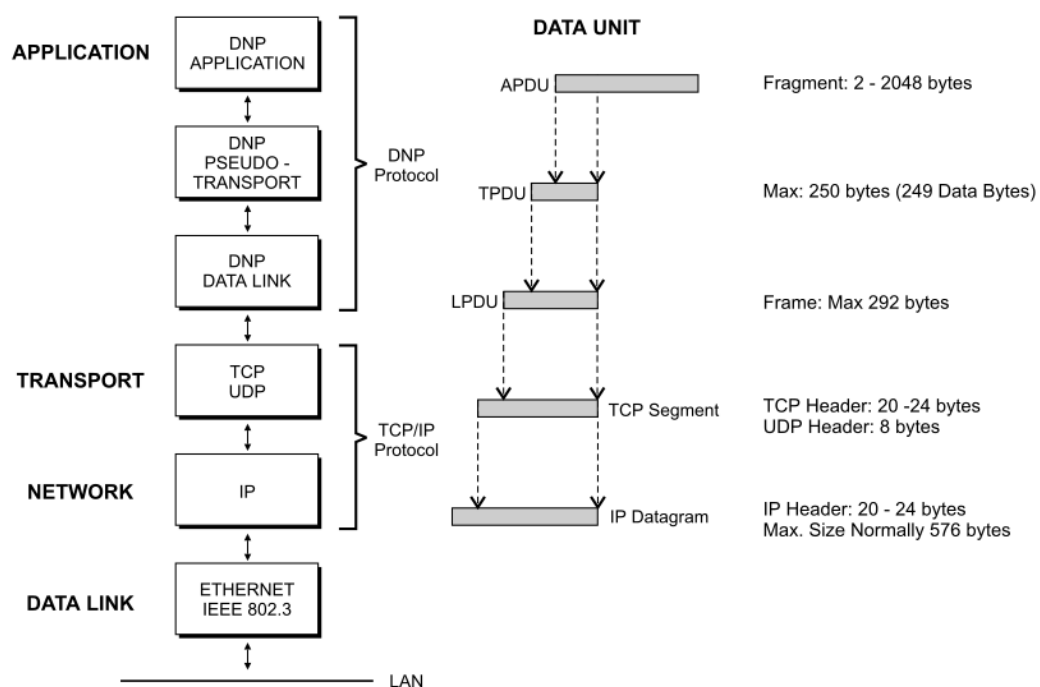


Figura 2.22 – Encapsulamento do protocolo DNP3 sobre TCP/IP.

Fonte: [Clarke e Reynolds \(2004\)](#).

O encapsulamento do DNP3 sobre TCP/IP é mostrado no diagrama da Figura 2.22, sendo que a construção das mensagens DNP3 para o TCP/IP são transparentes para as camadas do modelo EPA do DNP3 ([ORTEGA, 2015](#)), que é apresentado na Figura 2.23 para os dispositivos mestre e escravo.

A camada em destaque, denominada gerenciamento de conexão, é definida pela norma, sendo encarregada por estabelecer e encerrar as conexões TCP, transmitir, aceitar

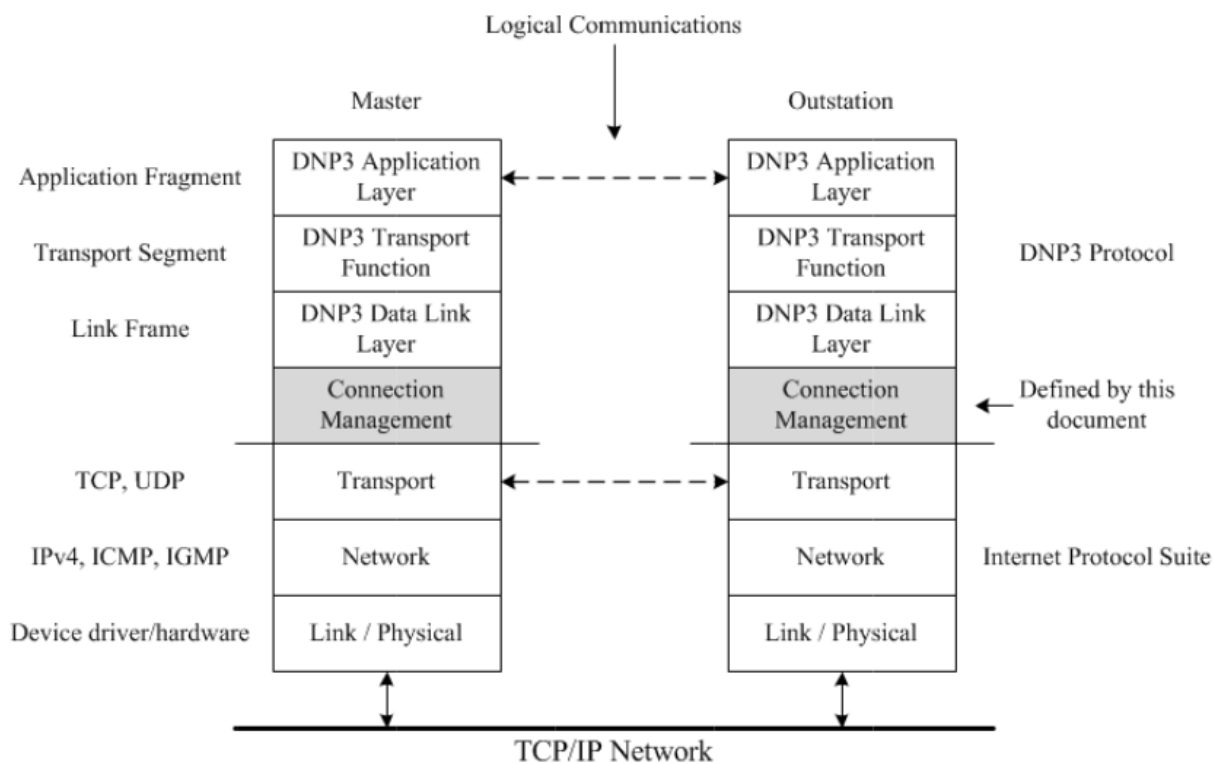


Figura 2.23 – Pilha de protocolos DNP3 sobre TCP/IP.

Fonte: [IEEE \(2012\)](#).

datagramas e enviar parte dos quadros DNP3 entre a camada de enlace do protocolo DNP3 ([ORTEGA, 2015](#)).

3 Desenvolvimento do trabalho

No desenvolvimento do trabalho, foi utilizado o *patch* gerador de tráfegos do protocolo DNP3 implementado por [Jaimes \(2012\)](#), utilizado por [Pereira \(2015\)](#) e [Ortega \(2015\)](#), baseado na classe do TCP já existente no NS-2, Application/TCPApp.

O emprego do *patch* de [Jaimes \(2012\)](#) envolveu alterações no núcleo do programa, com inclusão de parâmetro em alguns dos arquivos existentes, bem como na inclusão de outros arquivos, característicos da utilização do DNP3. Com o DNP3 devidamente implementado na versão do NS-2 utilizada, foi feita implementação similar para a comunicação PLC e por conseguinte a utilização conjunta do DNP3 e PLC, como pode ser visto nas seções que seguem.

3.1 Utilização do DNP3 no NS-2

[Jaimes \(2012\)](#) implementou com nível 1 o protocolo DNP3 na versão 2.33 do NS-2. Essa implementação foi adaptada para ser utilizada no presente trabalho, assim como foi feito por [Ortega \(2015\)](#) e [Pereira \(2015\)](#) em seus trabalhos. No presente trabalho, a simulação do protocolo foi utilizada na versão 2.35 do programa, compilada no sistema operacional Linux Ubuntu 16.04.2 LTS 64 bits.

Para implementação de um novo protocolo, é recomendado que seja seguida uma sequência, que envolve inclusive algumas alterações no núcleo do programa ([INSTITUTE, 2016](#)), como as seguintes:

- Definir estrutura de dados do protocolo;
- Executar ligação OTcl – C++;
- Alterar arquivos do núcleo do programa;
- Elaborar o código TCL.

3.1.1 Estrutura de dados do protocolo

A definição da estrutura de dados do protocolo inclui as “regras” que a mensagem deve seguir, sendo as do DNP3 apresentadas em [IEEE \(2012\)](#). Como explicado no Capítulo 2, o DNP3 possui uma estrutura do tipo mestre – escravo, podendo ser representado pelo diagrama de estado de cada estação, tendo na Figura 3.1 a representação do dispositivo escravo e na Figura 3.2 a representação da estação mestre.

Na Figura 3.1 pode-se observar o processo de inicialização (*Init*), no qual a estação está no estado ocioso (*Idle*), quando uma mensagem não solicitada é enviada via comando OTcl. Essa mensagem é transmitida pelo estado *Send Msg OTcl* que passa a aguardar uma resposta da estação mestre (*Waiting Resp from Master*). Caso a mensagem seja recebida, é retornado ao estado ocioso, porém, se o tempo de espera for atingido (*timeout*), é retornado ao estado de envio da mensagem OTcl e o ciclo se repete.

Já na Figura 3.2, de forma similar, a estação envia uma resposta à remota quando solicitada, ou envia uma solicitação de mensagem a ela, que pode ser gravação de dados, eventos de usuários, ficando no aguardo de dados da estação remota (*waiting for response*), sendo reenviada a mensagem quando o tempo de espera é atingido (*timeout*).

Com base no diagrama de estados, pôde-se elaborar os diagramas de classes. A estrutura do NS-2 é orientada a objetos, então seu núcleo é definido por meio de classes que seguem uma hierarquia, predominando a presença de estruturas que economizam código como é o caso da herança. No diagrama de classes desenvolvido por Jaimes (2012), foram implementadas a estação mestre e remota, como é ilustrado na Figura 3.3, na qual:

- a) **TcpAppMod:** Foi elaborada por Jaimes (2012) a partir de uma modificação da classe TcpApp já existente no NS-2. A opção por criar a classe ao invés de modificar a existente foi tomada para não causar alterações na distribuição oficial do programa. Esta classe simula o envio de dados a nível de camada de aplicação, por meio de uma sequência de tipo de fila de dados, cujas informações do pacote são armazenadas

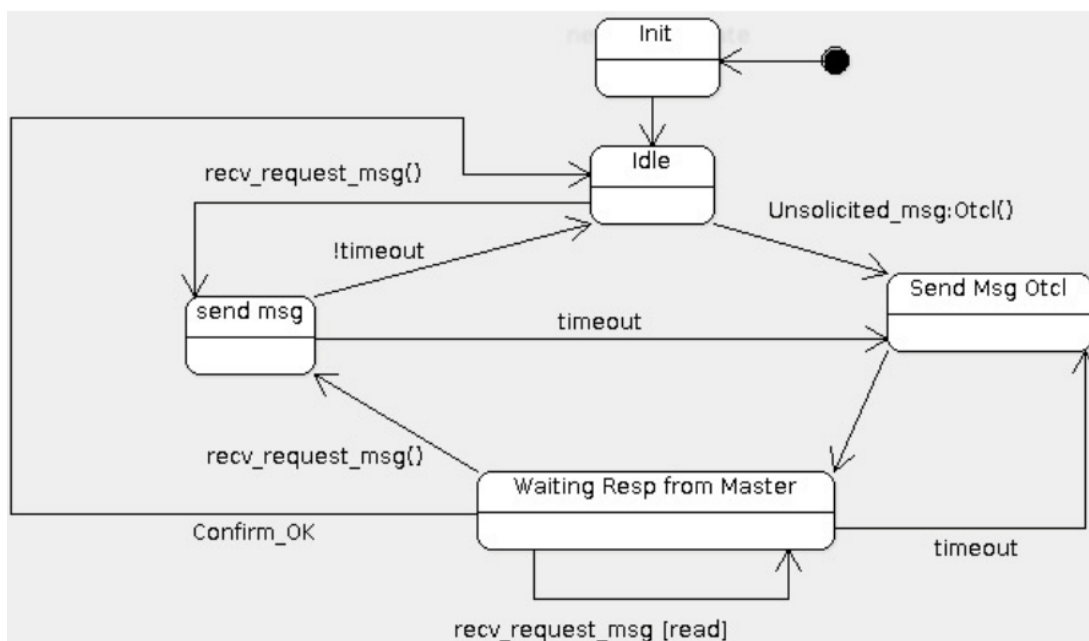


Figura 3.1 – Diagrama de estado da estação *Outstation*.

Fonte: Jaimes (2012).

para envio ao nó de destino.

b) **Dnp3App**: foi implementada a partir de uma classe de camada de aplicação também existente no programa, a classe HTTP, que permite trabalhar com várias ligações por parte de um agente a nível de aplicação. Dessa classe, outras duas herdam seus atributos, as classes `dnp3Appmaster` e `dnp3Appclient`, a primeira para a estação mestre e a segunda para estação escrava;

- A classe `dnp3Appmaster` utiliza três classes, a `stackid` que apresenta uma fila de dados que mantêm o valor de identificação de cada estação remota, a `stackdata` que apresenta a estrutura de dados de cada mensagem, para o caso de ser necessária uma retransmissão, e a `dnp3AppTimermaster`, que herda os atributos de `TimerHandler`, para implementar o temporizador do estado de espera.
- Já na classe `dnp3Appclient`, há apenas o `dnp3AppTimerclient`, cujo objetivo é o mesmo da temporização da classe mestre. Sua ativação é feita no momento em que a mensagem é acionada, sendo o seu tempo limite definido no arquivo `tcl` de simulação, que se atingido faz com que seja feito o reenvio de mensagem na estação destino (JAIMES, 2012).

Nas duas classes principais, `dnp3Appmaster` e `dnp3Appclient`, são herdados alguns métodos da classe `Dnp3App`, tais como *command* e *process_data*. Além destes, outros métodos são implementados nas suas classes correspondentes, como o *setmsg* e *resend*, cujo detalhamento é apresentado na Tabela 3.1.

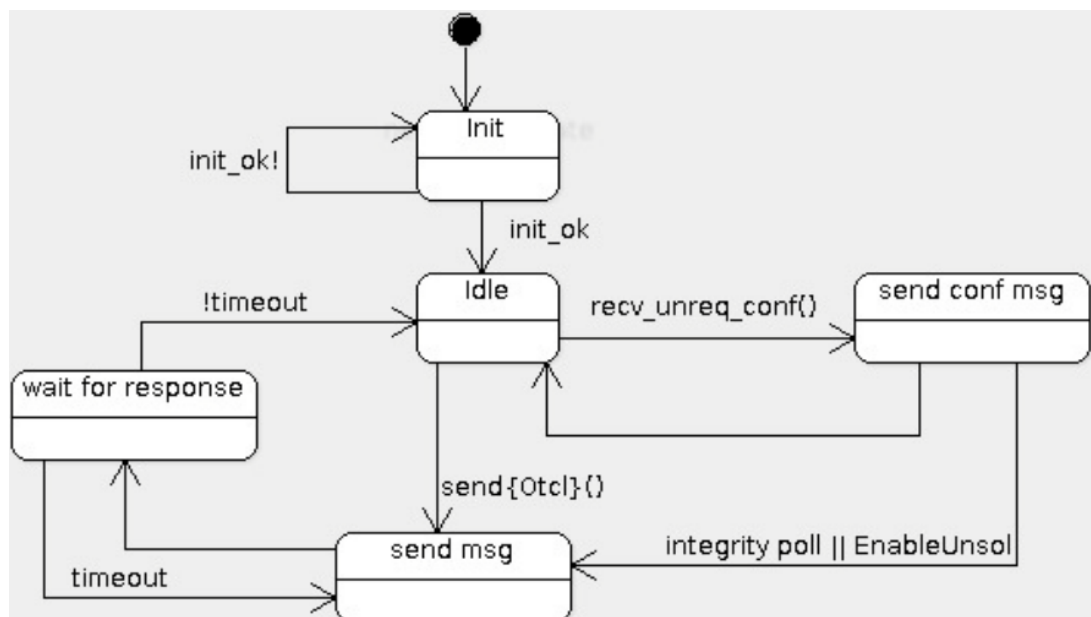


Figura 3.2 – Diagrama de estado da estação mestre.

Fonte: Jaimes (2012).

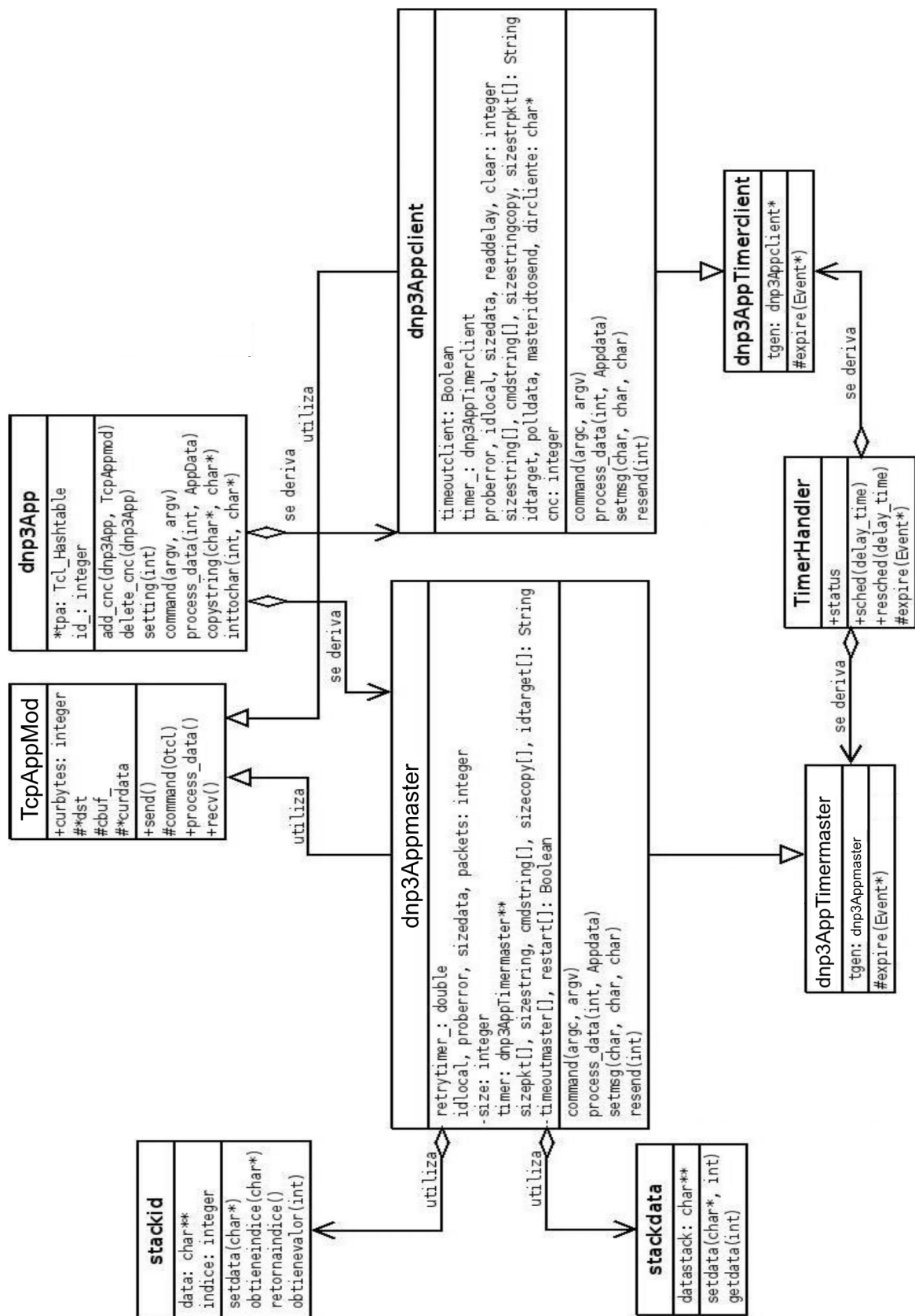


Figura 3.3 – Diagrama de classes.

Fonte: Ortega (2015 apud JAIMEs, 2012) – Adaptado.

Tabela 3.1 – Métodos das classes `dnp3AppMaster` e `dnp3AppClient`.

Método	Detalhes
<i>command</i>	Recebe comandos OTcl e configura variáveis para chamar o método <code>setmsg</code> , com o formato: <code><master> cmd <client> <bytes></code> . Nele é que são escolhidas as funções do DNP3 implementadas, “read” para leitura, “write” para escrita e “sel&op” para seleção e operação.
<i>process_data</i>	<p>É executado quando o nó recebe um pacote. Cada caso é diferente, mas o procedimento segue a sequência:</p> <ol style="list-style-type: none"> 1. Capturar o tamanho, comando e id da mensagem do cliente; 2. Cancelar o temporizador para parar a retransmissão; 3. Calcular o novo tamanho do pacote e enviar, se necessário. <p>A mensagem recebida do cliente tem o seguinte formato: <code><tamanho, comando, número do pacote, identificação do cliente ou do mestre></code></p>
<i>setmsg</i>	Cria e envia o formato da mensagem e define o temporizador caso a retransmissão seja necessária. O formato mensagem é: <code><tamanho, comando, número do pacote, identificação do <i>Master</i> ou do <i>Client</i>></code>
<i>resend</i>	É chamado quando o temporizador expira, isto é, o método “expire (Event *)” é executado na classe <code>dnp3AppTimermaster</code> . O método faz a retransmissão de mensagens chamando o método <code>setmsg</code> .

Com as definições passadas, foi criado no núcleo do programa uma pasta nomeada “dnp3”, contendo os arquivos `.h` e `.cc`, indicados a seguir:

- **tcpappmod.***: Modificada do arquivo original `tcpapp.*`, tem por função gerar o tráfego de dados e fazer a simulação das operações básicas, fazendo o encapsulamento do agente DNP3;
- **dnp3app.***: Define a classe `dnp3App`, que fornece os atributos para as outras duas, com construtores e destrutores;
- **dnp3appmaster.***: Define as funções descritas na Tabela 3.1 para a estação mestre;
- **dnp3appclient.***: Define as funções descritas na Tabela 3.1 para a estação escrava;

O sistema de mensagens utilizado apresenta a seguinte estrutura:

`<t.de sim><m/c:><recv/sended><cmd><pkt>`

em que:

- **t.sim**: Indica o tempo de simulação, dado pelo comando OTcl “[`$nsnow`]”;
- **m/c**: Indica que o evento pode ocorrer na estação mestre (m) ou na remota (c);

- **recv/sended:** O *recv* indica que o dado foi recebido e o *sended* que a mensagem foi enviada;
- **cmd:** Indica a função, que pode ser *Read*, *Write*, *Select and Operate*, *Restart* e *Event*;
- **pkt:** Indica o número de pacotes enviados, quando uma segmentação é necessária, devido ao tamanho do pacote.

3.1.2 Ligação OTcl – C++

Na Figura 2.6 foi mostrado um exemplo da ligação entre códigos OTcl e C++. Na implementação do protocolo, algumas ligações precisaram ser realizadas, como a mostrada na Figura 3.4, que apresenta o objeto “DNP3client”, utilizado na linguagem OTcl para criar uma estação remota, com o uso do comando “set client1 [new DNP3client 1]”, em que é criada a estação client1, interpretada pelo núcleo do programa escrito em C++.

Além do DNP3client, outras ligações foram necessárias na implementação do DNP3, como o “DNP3master” da classe dnp3Appmaster, “Application/TcpAppmod” da classe tcpappmod.

```
static class dnp3AppclientClass : public TclClass {
public:
    dnp3AppclientClass() : TclClass("DNP3client") {}
    TclObject* create(int argc, const char*const* argv) {
        if (argc != 5)
            return NULL;
        char *idclient = strdup(argv[4]);
        if (idclient == NULL)
            return NULL;
        char *client = strdup(argv[1]);
        return (new dnp3Appclient(idclient,client));
    }
} class_dnp3appclient_app;
```

Figura 3.4 – Ligação OTcl – C++ da classe dnp3client.

Além dessa ligação, foram feitas conexões que criaram vínculos entre classes OTcl e C++. A vinculação de variáveis é uma ferramenta que permite que programadores liguem uma variável de classe C++ a outra variável na classe OTcl vinculada, de modo que uma alteração em uma variável seja automaticamente refletida na outra. Dessa forma, uma variável OTcl, *iname*, pode ser ligada a uma variável C++, *cname*, por meio de uma das seguintes instruções no construtor de classe C++ (ISSARIYAKUL; HOSSAIN, 2011):

- `bind ("iname", &cname)` liga variáveis inteiras e reais.
- `bind_bw ("iname", &cname)` liga uma variável de largura de banda.
- `bind_time ("iname", &cname)` vincular uma variável de tempo.
- `bind_bool ("iname", &cname)` vincular uma variável booleana.

3.1.3 Alteração de arquivos do núcleo do programa

Alguns arquivos do núcleo do programa precisaram ser alterados, com o objetivo básico de informar ao NS-2 que um novo protocolo existe. Dentre os arquivos existentes, os que precisaram ser alterados foram:

- **Packet.h** – Nesse arquivo de cabeçalho disponível em `.../ns-allinone/ns-2.35/common/`, foi informada a existência do pacote do tipo DNP3, ilustrado na Figura 3.5.

```
// M-DART packets
static const packet_t PT_MDART = 72;

// DNP3 packets
static packet_t PT_DNP3 = 73;

// insert new packet types here
static packet_t PT_NTTYPE = 74; // This MUST be the LAST one
```

Figura 3.5 – Indicação do tipo de pacote PT_DNP3 no arquivo packet.h.

- **ns-default.tcl** – O ns-default está disponível em `.../ns-allinone/ns-2.35/tcl/lib/`, sendo responsável por várias definições, no DNP3 é utilizado para definir o valor padrão do tempo de retransmissão *retrytimer* para 1 segundo e da probabilidade de erros *probererror* para zero, como pode ser visto na Figura 3.6;

```
DNP3master set retrytimer_ 1
DNP3master set probererror_ 0
DNP3client set probererror_ 0
DNP3client set retrytimer_ 1
```

Figura 3.6 – Modificação do ns-default para o DNP3.

- **ns-packet.tcl** – Disponível em `.../ns-allinone/ns-2.35/tcl/lib/`, adiciona DNP3 ao *protolist*, ou seja, a lista de protocolos da camada de aplicação, como pode ser visto na Figura 3.7;

```
DCCP_CLOSEREQ # DCCP, transport protocol
# Application-Layer Protocols:
Message # a protocol to carry text messages
Ping # Ping
PBC # PBC
DNP3 # DNP3
# Wireless:
```

Figura 3.7 – Adicionando o DNP3 a lista de protocolos da camada de aplicação.

- **ns-process.h** – Disponível em `.../ns-allinone/ns-2.35/common/`, no arquivo é informada a existência do ADU – *Application Data Unit* (Unidade de dado de Aplicação) do tipo DNP3 ao programa, no “enumerador” AppDataType;

- **Makefile** – Toda mudança realizada no núcleo do programa deve ser novamente compilada por meio do Makefile, além disso as novas classes implementadas para o DNP3 precisam ser compiladas, para que sejam gerados os arquivos objeto .o, cuja existência deve ser indicada no Makefile.

3.1.4 O código TCL

Para utilização do NS-2, deve ser elaborado o código OTcl, as literaturas utilizadas sugerem inclusive a utilização dos arquivos disponíveis em <http://www.isi.edu/ns-allinone/ns-2.35/tcl/ex/>, como o simple.tcl. Pode-se ver um exemplo de arquivo tcl na Figura 3.8.

```
#Abre o arquivo de trace do NAM
set nf [open out.nam w]
$ns namtrace-all $nf

#Define o procedimento de finalização
proc finish {} {
    global ns nf
    $ns flush-trace
    #Fecha o arquivo de trace
    close $nf
    #Executa o nam no arquivo de trace
    exec nam out.nam &
    exit 0
}

#Cria três nós
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]

#Cria um link duplex entre os nós
$ns duplex-link $n0 $n1 1Mb 10ms DropTail

#Cria um agente UDP e associa ao nó n0
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0

# Cria uma fonte de tráfego CRB e junta ao agente udp0
set cbr0 [new Application/Traffic/CRB]
$nbr0 set packetSize_ 1000
$nbr0 set interval_ 0.01
$nbr0 attach-agent $udp0

#Cria um agente nulo (um 'traffic sink') e junta ao nó n1
set null0 [new Agent/Null]
$ns attach-agent $n1 $null0

#Conecta a fonte de tráfego ao agente nulo
$ns connect $nbr0 $null0

#Agenda eventos para o agente CBR
$ns at 0.5 "$nbr0 start"
$ns at 4.5 "$nbr0 stop"

#Chama o procedimento de finalização após 5s de simulação
$ns at 5.0 "finish"

#Inicia a simulação
$ns run
```

Figura 3.8 – Exemplo de código TCL.

Jaimes (2012) mostrou em seu trabalho a transmissão e retransmissão de mensagem em cenários ponto a ponto, utilizando TCP como protocolo de camada de transporte, mas com mudança no código padrão sugerido nos exemplos do NS-2, ao acrescentar:

- **Criação do gerador de tráfegos TCP com encapsulamento do Agente:** O comando `set encap [new Application/TcpAppmod $tcp1]` encapsula no Agente \$tcp1 previamente criado, no gerador de tráfegos encap, da classe TcpAppmod criada. Após isso o gerador de tráfegos encap é conectado a encap2 já que a comunicação é bidirecional, como pode ser observado na Figura 3.9.

```
set encap [new Application/TcpAppmod $tcp1]
set encap2 [new Application/TcpAppmod $tcp2]
$encap connect $encap2
$encap3 connect $encap4
```

Figura 3.9 – Gerador de tráfego e encapsulamento do agente.

- **Criação das estações mestre e escravo:** Ao criar as estações mestre e escravo(s), os tempos de retransmissão, com suas probabilidades de perda de pacote devem ser configurados, como é mostrado na Figura 3.10, em que se pode observar o tempo de retransmissão de 0,5 s e a probabilidade de erros de 0% para a estação mestre, enquanto na estação escrava o tempo de retransmissão de 1 s e a probabilidade de erros de 10%.

```
# Cria Estação Mestre
set master [new DNP3master]
$master set retrytimer_ 0.5
$master set proberror_ 0

# Cria Estação Escravo
set client1 [new DNP3client 1]
$client1 set retrytimer_ 1
$client1 set proberror_ 1
```

Figura 3.10 – Definição das estações.

- **Programação dos eventos da simulação:** Como mostrado na Figura 3.8, os eventos são agendados com o comando *\$ns at <tempo><procedimento>*. Para a simulação do DNP3, o procedimento é dado com a seguinte configuração: “<\$estação1> <evento> <\$estação2> <tamanho do pacote>”. No lugar de estação 1 e 2 são utilizadas as estações mestre e escravo(s), o evento aqui descrito pode ser *Read*, *Write*, *Select and Operate*, *Restart* e *Event*. O tamanho do pacote não é limitado pelo programa, tendo em vista que mensagens maiores que 250 bytes serão fragmentadas.

O DNP3 é implementado na camada de aplicação do modelo OSI, sendo TCP a opção escolhida para a camada de transporte, por garantir uma entrega confiável. Neste caso, torna-se desnecessária a verificação de erros por parte da camada de enlace do DNP3, podendo essa função ser desativada na integração.

3.2 Implementação do PLC

Assim como na implementação do DNP3, para o efetivo funcionamento do PLC no NS-2, vários arquivos do núcleo do programa precisaram ser estudados, acessados e algumas vezes alterados, dentre eles o *ns-mac.tcl*, no qual foi adicionado o valor padrão da variável de classe do Mac/PLC para o modo bidirecional, o *full_duplex_mode_*.

Sem essa adição, o programa apresenta mensagens de aviso como *Warning: no class variable...*, devido a estrutura organizacional do simulador. Na elaboração das classes referentes ao canal PLC e ao MAC – *Media Access Control* (Controle de Acesso ao Meio), alguns arquivos foram criados a partir de uma cópia existente, para manter a estrutura original do programa, como foi feito com o *TcpApp* para uso no DNP3, tais como:

- **mac-plc.h**, baseado em **simplemac.h**;
- **mac-plc.cc**, baseado em **simplemac.cc**;
- **channelmod.h**, baseado em **channel.h**;
- **channelmod.cc**, baseado em **channel.cc**.

A implementação do canal PLC e do controle MAC são descritos a seguir.

3.2.1 Modelagem do Canal

Pereira (2014) descreveu um canal genérico com BER – *Bit Error Rate* (Taxa de Erro de Bit) fixa, utilizando-a como parâmetro de entrada do sistema. Essa modelagem, porém, mostrou-se muito simplista, tendo em vista que atribuir uma BER fixa seria o mesmo que atribuir condições únicas para o funcionamento da rede PLC, com distâncias e impedâncias de rede com valores estáticos. Com base nessa premissa, modelou-se o canal PLC relacionando a BER com a SNR – *Signal Noise Rate* (Relação Sinal Ruído) presente no sinal, esta sim variando de acordo com as características construtivas da estrutura PLC.

A SNR passou então a ser dada como parâmetro de entrada do sistema, além das distâncias entre os nós que comprometem o atraso de propagação entre as estações mestre e remotas. Cada estação remota tem como entrada a SNR característica e a distância até a mestre. Tendo em vista que toda comunicação sempre é dada de ou para a estação mestre, fez-se necessário verificar dentro do canal qual estação remota estaria transmitindo ou recebendo o pacote. Para essa verificação, foi preciso primeiro identificar a origem e destino do dado, como ilustrado na Figura 3.11, em que se tem:

```
Packet *q;
hdr_cmn* ch = hdr_cmn::access(p);
int tam_pkt = ch->size();

hdr_ip* edr = hdr_ip::access(p);
Tcl::instance().evalf("puts \"Pacote: %d\"", ch->uid());
int tx_prob;
double snr_, PER;
ns_addr_t origem_ = edr->src();
int origem = origem_.addr_;
ns_addr_t destino_ = edr->dst();
int destino = destino_.addr_;
Tcl::instance().evalf("puts \"origem: %d\"", origem);
Tcl::instance().evalf("puts \"destino: %d\"", destino);
```

Figura 3.11 – Origem, destino e tamanho do pacote.

hdr_cmn: É a “estrutura” definida dentro de packet.h (disponível em .../ns-allinone/ns-2.35/common/) que possui os dados do pacote “p”. Dentre os dados do pacote, *size_* e *uid_* indicam o tamanho e a sua identificação, respectivamente. Por meio de

`ch->size()`, é acessado o tamanho do pacote para então ser atribuído a variável inteira criada (`tam_pkt`). Esse mesmo procedimento é realizado ao acessar `ch->uid()`, cujo valor é mostrado diretamente na tela de simulação, sem que seja necessária a atribuição do seu valor a uma variável local.

hdr_ip: É a “estrutura” definida dentro de `ip.h` (disponível em `.../ns-allinone/ns-2.35/common/`) que também possui dados do pacote “p”. Nessa, porém, `src_` e `dst_` indicam a origem e o destino do pacote, respectivamente. Esse acesso é realizado por meio de `edr->src()` e `edr->dst()`, atribuindo esses valores às variáveis `origem_` e `destino_`.

Essas variáveis de origem e destino são do tipo `ns_addr_t`, que é definida no arquivo `config.h` (disponível em `.../ns-allinone/ns-2.35/`) como uma “estrutura”, com a variável de endereço `addr_` do tipo inteiro de 32 bits. As variáveis inteiras origem e destino, criadas localmente, recebem seus valores acessando a variável `addr_` da estrutura `ns_addr_t` por meio de `origem_.addr_` para a origem do pacote e `destino_.addr_` para o seu destino.

Feita a identificação do endereçamento do pacote, e tendo como definição que à estação mestre sempre é dado o endereço “0”, é testado se a origem **ou** o destino têm o valor de 1 a N, em que “N” representa o número de estações que está sendo simulado, sendo atribuído à variável local `snr_` o valor da SNR correspondente à estação “X” selecionada, por meio de `snr_ = snrX_`.

A entrada dos dados de SNRs é feita por meio do arquivo OTcl, como apresentado na Subseção 3.1.4, sendo necessária também a ligação OTcl – C++, explicada na Subseção 3.1.2. Um trecho da ligação OTcl – C++ implementada no canal é ilustrada na Figura 3.12, com algumas das variáveis de SNRs e de distâncias, correspondentes ao número de estações remotas.

Já na Figura 3.13, é ilustrada a declaração de valores de SNRs e distâncias no código OTcl simulado, em que é atribuída à variável `snrX_` o valor de entrada (`$snrX`) dado pelo usuário e, de maneira similar, à variável `distX_` o valor de entrada `$distX`. Essa Figura apresenta ainda o canal *chan*, que corresponde a *Channel/PLCChannel*, cuja declaração é ilustrada na Figura 3.12.

A relação entre a SNR e a BER foi demonstrada por Augusto (2012) para a simulação do padrão PRIME, sendo apresentada na Tabela 3.2 para a modulação DQPSK¹ (Modulação Diferencial em Quadratura por Chaveamento de Fase, do inglês *Differential Quadrature Phase-Shift Keying*) e na Tabela 3.3 para a modulação DBPSK (Modulação Diferencial Binária por Chaveamento de Fase, do inglês *Differential Binary Phase-Shift Keying*). Por conseguinte, a PER (Taxa de Erro de Pacotes, do inglês *Packet Error Rate*) é

¹ Segundo Augusto (2012), no padrão PRIME, as taxas de transmissão das modulações DBPSK e DQPSK com codificador convolucional são baixas e correspondem a 21,4kbps e 42,9kbps, respectivamente, sendo o DBPSK mais robusto quanto a presença de ruídos.


```

static class PLCChannelClass : public TclClass {
public:
    PLCChannelClass() : TclClass("channel/PLCChannel") {}
    TclObject* create(int, const char*const*) {
        return (new PLCChannel);
    }
} class_PLC_channel;

static int PLCChannelIndex = 0;
PLCChannel::PLCChannel() : TclObject()
{
    (...)
    bind("snr1_", &snr1_);
    bind("snr2_", &snr2_);
    bind("snr3_", &snr3_);
    bind("snr4_", &snr4_);
    bind("snr5_", &snr5_);
    (...)
    bind("dist1_", &dist1_);
    bind("dist2_", &dist2_);
    bind("dist3_", &dist3_);
    bind("dist4_", &dist4_);
    bind("dist5_", &dist5_);
    (...)
}

```

Figura 3.12 – Ligação OTcl em channelmod.cc.

```

set snr1 8.0
set snr2 7.5
set snr3 7.0
set snr4 6.5
(...)
$opt(chan) set snr1_ $snr1
$opt(chan) set snr2_ $snr2
$opt(chan) set snr3_ $snr3
$opt(chan) set snr4_ $snr4
(...)
set dist1 100.0
set dist2 125.0
set dist3 150.0
set dist4 200.0
(...)
$opt(chan) set dist1_ $dist1
$opt(chan) set dist2_ $dist2
$opt(chan) set dist3_ $dist3
$opt(chan) set dist4_ $dist4
(...)

```

Figura 3.13 – Trecho do código OTcl da simulação elaborada.

mostrada na Equação 3.1:

$$PER = 1 - (1 - P_b)^N. \quad (3.1)$$

Tabela 3.2 – Relação entre SNR e BER na modulação DQPSK.

SNR (E_b/N_0) (dB)	BER
[0, 5[0,5
[5, $+\infty$ [$1,161 \times 10^4 \times e^{-2,011 \times x}$

Fonte: Augusto (2012).

Para utilização dessa modelagem, foi feita uma cópia dos arquivos *channel.** existentes no núcleo do programa, nomeados como *channelmod.h* e *channelmod.cc*, que implementam, dentro do método `recv(Packet* p, Handler * h)` da classe `PLCChannel`, o algoritmo mostrado nas Tabelas 3.2 e 3.3. O cálculo da PER, de acordo com o esquema de modulação escolhido, é ilustrado na Figura 3.14, em que se tem:

```

getchar();
if (modulation_ == 1){
    if (snr_ >= 0 && snr_ < 5.0){
        ber_ = 0.5;
    }else if (snr_ >= 5.0){
        ber_ = 1.161 * pow(10,4)*exp(-2.011*snr_);
    }
}else if (modulation_ == 2){
    if (snr_ >= 0 && snr_ < 5.0){
        ber_ = 0.5;
    }else if (snr_ >= 5.0){
        ber_ = 7.083 * pow(10,4)*exp(-2.372*snr_);
    }
}

PER = 1.0-pow((1.0-ber_),tam_pkt);

```

Figura 3.14 – Cálculo do PER de acordo com a BER.

`Tcl::instance().evalf("puts \"SNR...: Imprime na tela de prompt de comando o valor da SNR escolhida.`

`getchar()`: Interrompe a simulação, fazendo uma pausa até que o usuário pressione uma tecla. Isso é utilizado para que o programa pause toda vez que um pacote passa pelo canal, sendo necessário comentá-lo para simulações com grandes tráfegos de pacotes e/ou um número grande de nós.

if (modulation) == 1: Verifica o esquema de modulação escolhido via código OTcl, sendo 1 para a modulação DQPSK e 2 para a modulação DBPSK.

if (snr_ >= 0 && snr_ < 5.0): Testa se a SNR está no intervalo entre 0 e 5dB, atribuindo à BER o valor de 0,5. Para o caso da SNR com valores maiores ou iguais a 5.0 (**else if...**), a BER atribuída é mostrada na última célula da Tabela 3.2.

if (snr_ >= 0 && snr_ < 5.0): Testa se a SNR está no intervalo entre 0 e

Tabela 3.3 – Relação entre SNR e BER na modulação DBPSK.

SNR (E_b/N_0) (dB)	BER
[0, 5[0,5
[5, $+\infty$ [$7,083 \times 10^4 \times e^{-2,372 \times x}$

Fonte: Augusto (2012).

5dB, atribuindo à BER o valor de 0,5. Para o caso da SNR com valores maiores ou iguais a 5.0 (**else if...**), a BER atribuída é mostrada na última célula da Tabela 3.3.

PER = 1.0-pow...: Calcula por fim a taxa de erro de pacote, de acordo com a Equação 3.1, utilizando também para isso o tamanho do pacote que já é conhecido.

O método *recv(Packet* p, Handler * h)* recebe o pacote e o envia para a camada física, sendo efetuada a interrupção da transmissão de dados, para o caso do pacote ser perdido em função da PER calculada. Essa interrupção é feita baseada na utilização de uma função randômica, inerente do próprio programa chamada de *Random::uniform(0, 1)*, que gera um número aleatório com distribuição uniforme de 0 a 1.

O número aleatório gerado é testado com a PER, e para o caso da PER ser menor, o dado será transmitido chamando o método *sendUp(p, (Phy*)h)*, do contrário o pacote será perdido, sendo necessária sua retransmissão.

Embora não se tenha tratado sobre o atraso de propagação, ele pode ser definido como o atraso proveniente da propagação do sinal no meio físico (cabos da rede elétrica). Seu cálculo foi demonstrado por Wang et al. (2000) e utilizado por Augusto (2012), definido a partir da equação 3.2, em que:

$$\tau_p = \frac{d \times \sqrt{\epsilon_r}}{V_p} \quad (3.2)$$

- τ_p é o atraso de propagação;
- d é a distância entre as estações em metros;
- V_p é a velocidade da luz no vácuo em m/s;
- ϵ_r é a constante dielétrica do material isolante, que para o XLPE é igual a 2,3.

3.2.2 Controle de Acesso ao Meio

O Controle de Acesso ao Meio é a etapa da camada de enlace responsável por coordenar as transmissões de quadros para muitos nós (KUROSE; ROSS, 2013). O NS-2 possui uma série de classes implementadas que apresentam as características da subcamada MAC, de acordo com o meio de transmissão escolhido.

Essa coordenação das transmissões é dada por um controle de congestionamento, sendo necessária quando vários nós compartilham um único enlace, tendo em vista que os dados podem colidir ao concorrer com o uso do canal. Dentre os métodos de controle de acesso múltiplo, têm-se os protocolos de acesso controlado, de canalização e de acesso aleatório que são apresentados na Figura 3.15.

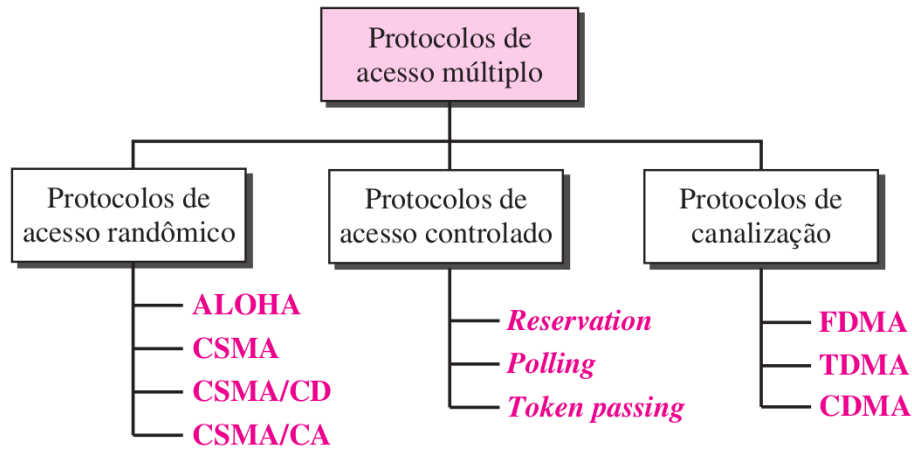


Figura 3.15 – Taxonomia dos protocolos de acesso múltiplo.

Fonte: Forouzan (2009).

Um exemplo de implementação da subcamada MAC no NS-2 é dada por Jin e Kunz (2010), que a fez para o padrão HomePlug C&C (PLC). Em seu trabalho, ele se baseou na pilha de protocolos sem fio IEEE 802.11, que apesar de apresentar o modelo de propagação de rádio também característico do PLC, mostrou-se bastante generalista em sua implementação, tendo em vista que deve-se configurar elementos não existentes numa rede PLC como tipo de antena, além de forçar a velocidade de transmissão para parâmetros próximos ao desejado.

Para utilização de um controle de congestionamento neste trabalho, mostrou-se necessária a implementação de um dos protocolos de acesso aleatórios, em que nenhuma estação é superior. Essa indicação é feita também pela literatura para o uso no PLC, como é o caso de IEEE (2013), que sugere como método de controle de colisão para comunicação PLC banda estreita uma variação do CSMA – *Carrier Sense Multiple Access* (Acesso Múltiplo de Detecção de Portadora), o CSMA/CA que apresenta como diferença o *Collision Avoidance*, ou seja, mecanismos para evitar colisão.

Com isso, a subcamada MAC implementada teve como objetivo atender as duas situações possíveis, quando não é necessário o controle de congestionamento, a comunicação ponto a ponto e, quando é necessário, a comunicação multiponto.

3.2.2.1 Comunicação ponto a ponto

Na comunicação ponto a ponto, foi utilizado um controle de acesso ao meio simples, sem controle de congestionamento, para a comunicação com apenas dois nós (ou mais, desde que haja o enlace de comunicação direto), já que neste caso não há concorrência de uso do canal e, segundo Forouzan (2009), as funcionalidades de detecção de portadora e de colisões da subcamada MAC podem ser desativadas.

Para realizar simulações do tipo ponto a ponto, foi utilizado como MAC a classe **Mac** já existente nos arquivos `mac.h` e `mac.cc` (disponível em `.../ns-allinone/ns-2.35/mac/`), que herda seus atributos da classe *BiConnector*, sendo a classe mãe de todas as outras desse tipo, como a *Mac/Sat* para satélites ou a *Mac/802_3* para Ethernet. A *Mac* padrão faz apenas o encaminhamento do pacote recebido para as camadas superior ou inferior, por meio dos métodos *recv*, *sendUp* e *sendDown*, como pode ser observado na Figura 3.16, que ilustra o início da estrutura de dados de *Mac*.

```
class Mac : public BiConnector {
public:
    Mac();
    virtual void recv(Packet* p, Handler* h);
    virtual void sendDown(Packet* p);
    virtual void sendUp(Packet *p);
    virtual void resume(Packet* p = 0);
};
```

Figura 3.16 – Declaração de Métodos na classe MAC.

3.2.2.2 Comunicação multiponto

Na comunicação multiponto, devido à necessidade de controle de congestionamento, foi utilizado para criação da classe *MacPLC* a *MacSimple*, que é uma estrutura de controle de acesso ao meio também utilizada na comunicação sem fio. Para isso, foram criados os arquivos `mac-plc.h` e `mac-plc.cc` dentro da pasta *PLC*, no núcleo do programa, cuja implementação contou com a mesma sequência descrita anteriormente para a ligação OTcl – C++ ocorrida no DNP3 e PLC.

Os arquivos criados foram baseados no `mac-simple.*` (disponível em `<.../ns-allinone/ns-2.35/mac/>`), contando nesse caso com um controle de congestionamento baseado no CSMA, que, de forma simplificada, apresenta um tempo aleatório para “escutar” novamente o canal antes de tentar a retransmissão, para o caso do canal estar ocupado.

Para entender melhor o que foi implementado, pode-se ver o diagrama de classes mostrado na Figura 3.17, em que se tem:

- **MacPLC**: Construída por meio de uma modificação dos arquivos `mac-simple.*`, herda os atributos da classe *Mac* e faz outras implementações, em destaque o tratamento para os casos de congestionamento.
- **BackoffTimer**: *Friend Class* de *MacPLC*, é chamada quando o estado de recebimento do canal (`rx_state_`) não está livre, por meio de `trace_event(“BACKING_OFF”,p)`.
- **MacPLCTimer**: Implementada dentro dos arquivos `mac-plc.*`, herda os atributos da classe *Handler*, tem como funcionalidade as operações de controle de tempo que

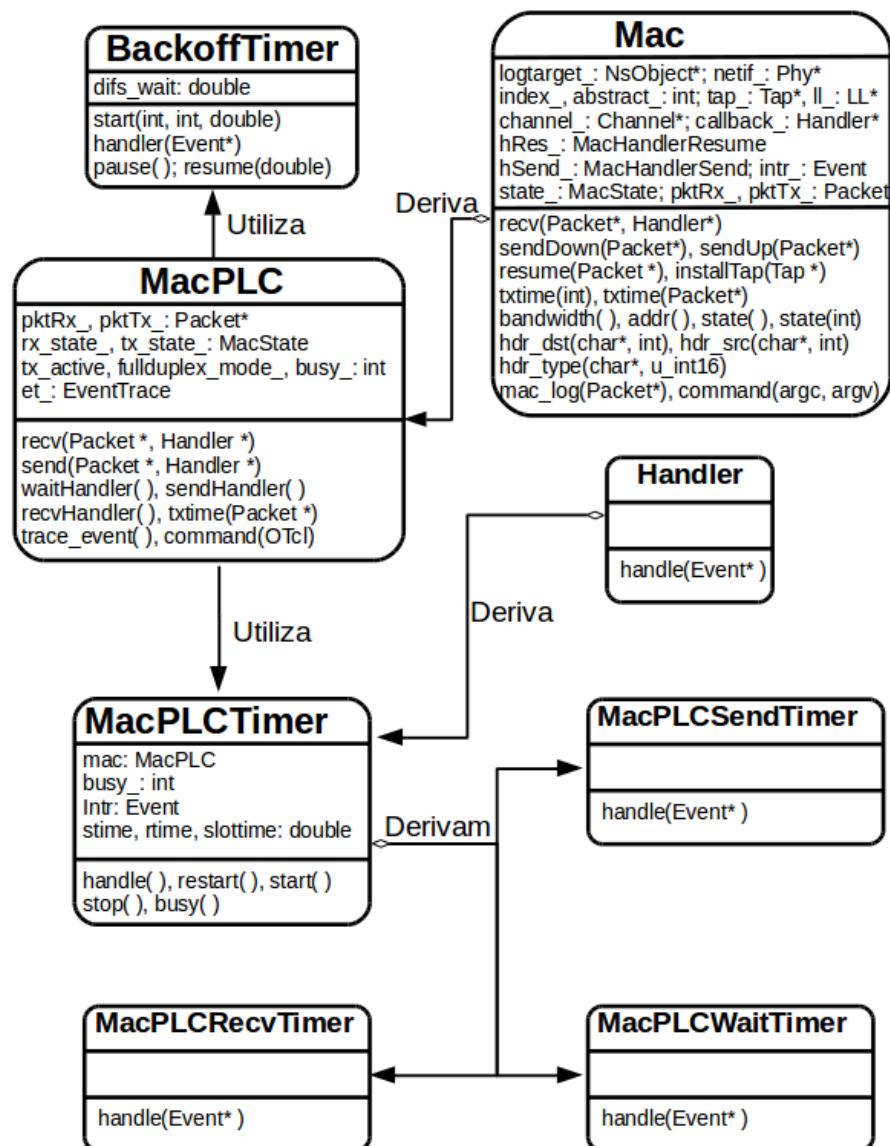


Figura 3.17 – Diagrama de classes da Subcamada Mac.

são utilizadas pela classe MacPLC, da qual também derivam outras três subclasses de tempo.

- **MacPLCWaitTimer**: Tem como função atrasar o envio de pacotes.
- **MacPLCSendTimer**: Responsável por gerenciar o tempo para finalizar o envio de pacotes.
- **MacPLCRecvTimer**: Responsável por gerenciar o tempo para finalizar o recebimento de pacotes.

3.2.3 Validação da implementação

Na validação foi realizado o envio de dados em dois cenários, uma comunicação ponto a ponto e outra multiponto, comparando os resultados aos encontrados por [Ortega e Shinoda \(2013\)](#), tomando como parâmetros de entrada os valores apresentados em seu trabalho.

Para o primeiro cenário, a probabilidade de erro foi igual a 1 (10%), o tempo de retransmissão de 1 s para a estação remota e 0,5 s para a estação mestre, largura de banda de 100Mbps, com atraso de 0,6ms. O envio de pacotes foi realizado com tamanhos diversos, na função de leitura, para os pacotes partindo da estação mestre e mensagem não solicitada partindo da estação remota. Os resultados são apresentados na Figura 3.18.

```
0.5 OutStation 1: 1, msg <event> sended, pkt 1, size 79
0.50060952000000003 Master: rcv <event> from OutStation _o43 numberpkt(1)
0.50060952000000003 Master: msg <event_ok> sended to OutStation _o43
0.50121272000000006 OutStation 1: rcv <event_ok>
Master: Active Timer 0.000000 Seg.
1 Master: 9, msg <read> sended to OutStation _o43 (pkt 1)
1.00061584 OutStation 1: rcv <read>, size 130
1.00061584 OutStation 1: msg <read_ok> sended, pkt 1
1.00123776 Master: rcv <read_ok> from OutStation _o43 numberpkt(1)
Master: Disable the Timer 1
1.00123776 Master: msg <conf_read> sended to OutStation _o43
1.00184176 OutStation 1: rcv <conf_read>, size 10
1.5 OutStation 1: 1, msg <event> sended, pkt 1, size 44
1.50060671999999999 Master: rcv <event> from OutStation _o43 numberpkt(1)
1.50060671999999999 Master: msg <event_ok> sended to OutStation _o43
1.50122032000000001 OutStation 1: rcv <event_ok>
```

Figura 3.18 – Ponto a ponto – validação.

Comparando aos resultados apresentado na Figura 3.19, nota-se que o tempo de envio e recebimento apresentado é semelhante, não indicando variações, mesmo com alteração da *seed*.

Tempo (s)	Nó	Tipo	Comando	PKT.	Tam. (b)
0.5	OutStation1	sended	event	1	79
0.500609	Master	received	event	1	79
0.500609	Master	sended	event_ok	1	0
0.501212	OutStation1	receives	event_ok	1	0
1.0	Master	active timer	0.5 s	0	0
1.0	Master	sended	read	1	130
1.0006	OutStation1	received	read	1	130
1.0006	OutStation1	sended	read_ok	1	0
1.001239	Master	received	read_ok	1	0
1.001239	Master	disable timer	1 s	0	0
1.001239	Master	sended	conf_read	1	10
1.001843	OutStation1	received	conf_read	1	10
1.5	OutStation1	sended	event	1	44
1.500606	Master	received	event	1	44
1.500606	Master	sended	event_ok	1	0
1.501220	OutStation1	received	event_ok	1	0

Figura 3.19 – Mensagem não solicitada e função de leitura.

Fonte: [Ortega e Shinoda \(2013\)](#).

No segundo cenário é apresentada uma configuração multiponto com 2 estações remotas ligadas à estação mestre, diferindo da comunicação ponto a ponto no atraso,

com uma das estações apresentando 1ms para esse parâmetro, sendo feito o envio de pacotes de leitura e escrita provenientes da estação mestre. Os resultados da simulação são apresentados na Figura 3.20, que podem ser comparados aos resultados de Ortega e Shinoda (2013) apresentados na Figura 3.21.

```
0.5 OutStation 1: 3, msg <event> sent, pkt 1, size 79
[0.50061] _o66 receives a packet but no callback!
0.5100000000000001 OutStation 2: 8, msg <event> sent, pkt 1, size 67
0.5106085599999999 Master: rcv <event> from OutStation _o71 numberpkt(1)
0.5106085599999999 Master: msg <event_ok> sent to OutStation _o71
Master: Active Timer 0.000000 Seg.
1 Master: 6, msg <read> sent to OutStation _o70 (pkt 1)
1 OutStation 1: timeout, resend
1 OutStation 1: 3, msg <event> sent, pkt 1, size 79
[1.00061] _o66 receives a packet but no callback!
1.01 OutStation 2: timeout, resend
1.01 OutStation 2: 8, msg <event> sent, pkt 1, size 67
1.0106085600000001 Master: rcv <event> from OutStation _o71 numberpkt(1)
1.0106085600000001 Master: msg <event_ok> sent to OutStation _o71
1.1000000000000001 OutStation 2: 0, msg <event> not sent
1.5 OutStation 1: 6, msg <event> sent, pkt 1, size 39
[1.50061] _o66 receives a packet but no callback!
Master: Active Timer 0.000000 Seg.
1.6000000000000001 Master: 1, msg <read> sent to OutStation _o70 (pkt 1)
1.6000000000000001 OutStation 2: timeout, resend
1.6000000000000001 OutStation 2: 8, msg <event> sent, pkt 1, size 92
1.60061056 Master: rcv <event> from OutStation _o71 numberpkt(1)
1.60061056 Master: msg <event_ok> sent to OutStation _o71
Master: Active Timer 0.000000 Seg.
1.6020380000000001 Master: 1, msg <write> sent to OutStation _o71 (pkt 1)
2 OutStation 1: timeout, resend
```

Figura 3.20 – Multiponto – validação.

Tempo (s)	Nó	Tipo	Comando	PKT.	Tam. (b)						
0.5	Outstation1	sended	event	1	79	1.101010	Master	sended	event_ok	1	0
0.500609	Master	received	event	1	79	1.102024	Outstation2	received	event_ok	1	0
0.500609	Master	sended	event_ok	1	0	1.500000	Outstation1	sended	event	1	39
0.501212	Outstation1	received	event_ok	1	0	1.500606	Master	received	event	1	39
0.510000	Outstation2	sended	event	1	67	1.500606	master	sended	event_ok	1	0
0.510085	Master	received	event	1	67	1.501219	Outstation1	received	event_ok	1	0
0.510085	Master	sended	event_ok	1	0	1.600000	Master	timeout	resend	1	0
0.512011	Outstation2	received	event_ok	1	0	1.600000	Master	active timer	0.5 s	0	0
1.000000	Master	active timer	0.5 s	0	0	1.600000	Master	sended	read	1	130
1.000000	Master	sended	read	1	71	1.601015	Outstation2	received	read	1	130
1.000610	Outstation1	received	read	1	71	1.601015	Outstation2	sended	read_ok	1	0
1.000610	Outstation1	sended	read_ok	1	0	1.602038	Master	received	read_ok	1	0
1.001222	Master	received	read_ok	1	0	1.602038	Master	disable time	1 s	0	0
1.001222	Master	disable timer	1 s	0	0	1.602038	Master	sended	conf_read	1	10
1.001222	Master	sended	conf_read	1	10	1.602038	Master	active time	0.5 s	0	0
1.001826	Outstation1	received	conf_read	1	10	1.602038	Master	sended	write	1	17
1.100000	Master	active timer	0.5 s	0	0	1.603042	Outstation2	received	conf_read	1	10
1.100000	Master	not sended	read	1	0	1.603048	Outstation2	received	write	1	17
1.100000	Outstation2	sended	event	1	92	1.603048	Outstation2	sended	write_ok	1	17
1.101010	Master	received	event	1	92	1.604053	Master	received	write_ok	1	0
						1.604053	Master	disable timer	2	0	0

Figura 3.21 – Simulação Multiponto.

Fonte: Ortega e Shinoda (2013).

Novamente os valores encontrados apresentaram-se coerentes com o modelo de base, validando assim o seu uso.

4 Simulações e Resultados

Neste Capítulo são apresentadas as simulações e resultados deste trabalho. Na Seção 4.1, são apresentados resultados de simulações para validar o DNP3 proposto por Jaimes (2012). Na Seção 4.2, é validado o DNP3 encapsulado em TCP/IP sobre PLC, com simulação de alguns cenários característicos.

4.1 Simulação do DNP3

A versão implementada do DNP3 no NS-2 foi descrita na Seção 3.1 e, para validar seu uso, algumas situações foram simuladas, cujos resultados podem ser vistos a seguir.

4.1.1 Envio das funções básicas do DNP3

Dentre as funções básicas do DNP3, tem-se a solicitação de leitura, escrita, seleção e operação pelo mestre, o envio de mensagem não solicitada por parte da estação remota, além da função *restart*, que testa todas as funções de leitura e escrita entre as estações.

No elaboração do Código OTel para as simulações das funções básicas, foram utilizadas as configurações a seguir, assim como as mostradas na Figura 4.1.

- **\$defaultRNG seed 150:** Número aleatório que deve ser alterado para não repetir sempre os mesmos resultados¹;
- **Número de estações:** Duas, uma mestre e outra escrava;
- **Enlace do nó, largura de banda, atraso e regra de fila:** *duplex-link*, 10Mb, 1ms, *Droptail*;
- **Probabilidade de perda de pacotes:** Mestre = 20%, estação remota = 20%;
- **Tempo de retransmissão:** Mestre = 1s, estação remota = 0,5s.

Os dados da Figura 4.1 podem ser melhor compreendidos com auxílio da Tabela 4.1, que indica o instante de tempo de cada evento, com sua estação de origem e tamanho. Ao simular, são apresentados resultados na tela do *prompt* de comando, que mostram o decorrer da transmissão dos pacotes, conforme são enviados, recebidos ou perdidos, como pode ser visto na Figura 4.2.

¹ O NS-2 utiliza um algoritmo para elaboração de números (pseudo)aleatórios, de modo que, ao alterar o número padrão (que inicialmente é 1), os números gerados serão diferentes, do contrário não haveriam variações nos resultados.

```

#          Simulação 1 - Envio simples das funções básicas
$ns at 0.1 "$client1 event $master 251"
$ns at 3 "$master write $client1 200"
$ns at 10 "$master read $client1 200"
$ns at 20 "$master sel&op $client1 300"
$ns at 25 "$client1 restart $master 100"
#          Simulação 2 - Envio cíclico de msg não solicitada
for {set i 30} {$i < 130} {set i [expr $i*0.1]} {
$ns at $i "$client1 event $master 100"
}
# Configura o momento de finalização da simulação
$ns at 150 "fim"
# Inicia a Simulação
$ns run

```

Figura 4.1 – Trecho do código OTcl para simulação do DNP3.

```

0.100000000000000001 OutStation 1: 1, msg <event> not sent
0.59999999999999998 OutStation 1: timeout, resend
0.59999999999999998 OutStation 1: 8, msg <event> sent, pkt 2, size 292
0.59999999999999998 OutStation 1: 8, msg <event> sent, pkt 1, size 13
0.60130799999999995 Master: rcv <event> from OutStation _o43 numberpkt(1)
0.60130799999999995 Master: msg <event_ok> sent to OutStation _o43
0.60233999999999999 OutStation 1: rcv <event_ok>
Master: Active Timer 0.000000 Seg.
3 Master: 4, msg <write> sent to OutStation _o43 (pkt 1)
3.0012208 OutStation 1: rcv <write>, size 200
3.0012208 OutStation 1: msg <write_ok> sent, size 17
3.00226639999999999 Master: rcv <write_ok> from OutStation _o43 numberpkt(1)
Master: Disable the Timer 1
Master: Active Timer 0.000000 Seg.
10 Master: 1, msg <read> not sent to OutStation _o43
11 Master: timeout timer 1, resend
Master: Active Timer 0.000000 Seg.
11 Master: 8, msg <read> sent to OutStation _o43 (pkt 1)
11.0012208 OutStation 1: rcv <read>, size 200
11.0012208 OutStation 1: msg <read_ok> sent, pkt 2
11.0012208 OutStation 1: msg <read_ok> sent, pkt 1
11.0025632000000001 Master: rcv <read_ok> from OutStation _o43 numberpkt(1)
Master: Disable the Timer 1
11.0025632000000001 Master: msg <conf_read> sent to OutStation _o43
11.0036032000000001 OutStation 1: rcv <conf_read>, size 10

```

Figura 4.2 – Mensagem de saída no *prompt* de comando.

Da Figura 4.1 e Tabela 4.1, no instante 0,1s é projetado o envio de uma mensagem não solicitada por parte da estação remota, porém o envio não é realizado, o que pode ser visto na Figura 4.2. O tempo limite *timeout* é atingido após decorridos 500ms, devido ao número aleatório gerado no núcleo do programa estar dentro da probabilidade de perda de pacotes de 20% proposta. Já que o pacote não foi transmitido, é feita uma nova tentativa de envio da mensagem (*resend*), quando finalmente o pacote consegue ser transmitido.

Essa mensagem, que originalmente tinha 251 bytes, excede o tamanho máximo indicado para o TPDU de 250 bytes, sendo dividido em dois pacotes, um com 250 bytes e outro com 1 byte. Ele é apresentado com 13 bytes por conter 10 bytes da camada de enlace do DNP3 e 2 bytes do CRC. Essa transmissão de dados encerra-se no instante 0.60233999999999999s com o recebimento pela estação remota, do pacote de confirmação

Tabela 4.1 – Envio das funções básicas do DNP3.

Tempo (s)	Evento	Estação de Origem	Tamanho (bytes)
0,1	Mensagem não solicitada	Escravo	251
3,0	Escrita	Mestre	200
10	Leitura	Mestre	200
20	Seleção e Operação	Mestre	300
25	Reinício	Escravo	100
30-130	Mensagem não solicitada	Escravo	100

de envio, proveniente da estação mestre.

Ainda na Figura 4.2, são mostrados os eventos nos instantes 3 e 10s. No primeiro um evento é reportado sem problemas, a escrita de dados por parte da estação mestre, com a indicação de um pacote de confirmação com 17 bytes, finalizando a transmissão no instante 3,0022663999999999s.

Foi necessária a retransmissão do evento do instante 10s, desta vez 1s depois. A mensagem de leitura enviada pela estação mestre teve tamanho de 200 bytes, porém a mensagem de retorno foi gerada de acordo com as informações existentes na estação remota. No programa foi gerado um tamanho de dados aleatório, maior do que o enviado pela estação mestre. Pela Figura 4.2, pode-se ver apenas que a mensagem foi superior a 250 bytes e inferior a 500 bytes, já que foram gerados 2 pacotes. O valor exato do pacote pode ser verificado na Figura 4.3, que foi de 290 bytes, já que o último apresentou 96 bytes, sendo 40 do protocolo TCP/IP, 10 da camada de enlace do protocolo, e 6 de CRC.

```

1 + 0.6      1 0 DNP3 332 ----- 0 1.0 0.0 -1 0    21 + 11.001221 1 0 DNP3 96 ----- 0 1.0 0.0 -1 7
2 - 0.6      1 0 DNP3 332 ----- 0 1.0 0.0 -1 0    22 - 11.001486 1 0 DNP3 96 ----- 0 1.0 0.0 -1 7
3 + 0.6      1 0 DNP3 53  ----- 0 1.0 0.0 -1 1    23 r 11.002486 1 0 DNP3 332 ----- 0 1.0 0.0 -1 6
4 - 0.600266 1 0 DNP3 53  ----- 0 1.0 0.0 -1 1    24 r 11.002563 1 0 DNP3 96 ----- 0 1.0 0.0 -1 7
5 r 0.601266 1 0 DNP3 332 ----- 0 1.0 0.0 -1 0    25 + 11.002563 0 1 DNP3 50 ----- 0 0.0 1.0 -1 8
6 r 0.601308 1 0 DNP3 53  ----- 0 1.0 0.0 -1 1    26 - 11.002563 0 1 DNP3 50 ----- 0 0.0 1.0 -1 8
7 + 0.601308 0 1 DNP3 40 ----- 0 0.0 1.0 -1 2    27 r 11.003603 0 1 DNP3 50 ----- 0 0.0 1.0 -1 8
8 - 0.601308 0 1 DNP3 40 ----- 0 0.0 1.0 -1 2    28 + 20          0 1 DNP3 205 ----- 0 0.0 1.0 -1 9
9 r 0.60234  0 1 DNP3 40 ----- 0 0.0 1.0 -1 2    29 - 20          0 1 DNP3 205 ----- 0 0.0 1.0 -1 9
10 + 3        0 1 DNP3 276 ----- 0 0.0 1.0 -1 3    30 r 20.001164 0 1 DNP3 205 ----- 0 0.0 1.0 -1 9
11 - 3        0 1 DNP3 276 ----- 0 0.0 1.0 -1 3    31 + 20.001164 1 0 DNP3 205 ----- 0 1.0 0.0 -1 10
12 r 3.001221 0 1 DNP3 276 ----- 0 0.0 1.0 -1 3    32 - 20.001164 1 0 DNP3 205 ----- 0 1.0 0.0 -1 10
13 + 3.001221 1 0 DNP3 57 ----- 0 1.0 0.0 -1 4    33 r 20.002328 1 0 DNP3 205 ----- 0 1.0 0.0 -1 10
14 - 3.001221 1 0 DNP3 57 ----- 0 1.0 0.0 -1 4    34 + 20.002328 0 1 DNP3 205 ----- 0 0.0 1.0 -1 11
15 r 3.002266 1 0 DNP3 57 ----- 0 1.0 0.0 -1 4    35 - 20.002328 0 1 DNP3 205 ----- 0 0.0 1.0 -1 11
16 + 11       0 1 DNP3 276 ----- 0 0.0 1.0 -1 5    36 r 20.003492 0 1 DNP3 205 ----- 0 0.0 1.0 -1 11
17 - 11       0 1 DNP3 276 ----- 0 0.0 1.0 -1 5    37 + 20.003492 1 0 DNP3 205 ----- 0 1.0 0.0 -1 12
18 r 11.001221 0 1 DNP3 276 ----- 0 0.0 1.0 -1 5    38 - 20.003492 1 0 DNP3 205 ----- 0 1.0 0.0 -1 12
19 + 11.001221 1 0 DNP3 332 ----- 0 1.0 0.0 -1 6    39 r 20.004656 1 0 DNP3 205 ----- 0 1.0 0.0 -1 12
20 - 11.001221 1 0 DNP3 332 ----- 0 1.0 0.0 -1 6

```

Figura 4.3 – Arquivo na simulação do DNP3.

Segundo Ortega (2015), quando a estação remota está configurada para enviar uma mensagem não solicitada e entra no estado de *reset*, ela deve enviar mensagens não solicitadas sem dados para a estação mestre, até receber confirmação, quando então é enviado o pacote que se deseja transmitir, não sendo especificado esse número de retransmissões. Esta função de reinício é apresentada na Figura 4.4.

```

25 OutStation 1: 0, msg <restart> not sent
25.5 OutStation 1: timeout, resend
25.5 OutStation 1: 0, msg <restart> not sent
26 OutStation 1: timeout, resend
26 OutStation 1: 9, msg <restart> sent, pkt 1, size 31
26.001056800000001 Master: rcv <restart> from OutStation _o43 numberpkt(1)
26.001056800000001 Master: msg <clear> sent
  Master: Active Timer 0.000000 Seg.
26.001056800000001 Master: 5, msg <read> sent to OutStation _o43 (pkt 1)
26.0021024000000002 OutStation 1: rcv <clear>, size 17
26.0021592000000001 OutStation 1: rcv <read> integrity poll
26.0021592000000001 OutStation 1: msg <read_ok> integrity poll sent
26.0032712 Master: rcv <read_ok> from OutStation _o43 numberpkt(1)
  Master: Disable the Timer 1
26.0032712 Master: msg <conf_read> sent to OutStation _o43
  Master: Active Timer 0.000000 Seg.
26.0032712 Master: 2, msg <write> sent to OutStation _o43 (pkt 1)
26.0043112 OutStation 1: rcv <conf_read>, size 10
26.004367999999999 OutStation 1: rcv <write - Enable unsol msg>
26.004367999999999 OutStation 1: msg <write_ok Enable unsol msg> sent, size 17

```

Figura 4.4 – Mensagem de saída da função *reset* no *prompt* de comando.

Pode-se ver ainda a continuidade desta função na Figura 4.4, em que a estação remota recebe o pacote denominado *integrity poll* proveniente do mestre, um comando de leitura do estado do dispositivo e, caso necessário, é feito o envio do comando de escrita (instante 26,0032712s) para estabelecer seus valores de entradas / saídas.

Como último item simulado, foi configurado o envio cíclico de mensagens não solicitadas a cada 1s, iniciando no instante 30 s e finalizando no instante 130 s, sendo esperado o envio de 100 mensagens de 100 bytes. Para essa etapa de simulação, foi ajustado o tempo de retransmissão para 200 ms, para ambas as estações, variada a probabilidade de perda de pacote com os valores de 20, 30 e 40%, também das duas estações, alternando o valor do *seed* que altera o algoritmo do gerador de números aleatórios do núcleo do programa para evitar que repita sempre o mesmo resultado. O resultado dessa etapa da simulação é apresentado na Tabela 4.2.

Tabela 4.2 – Resultado do envio cíclico das mensagens não solicitadas.

Prob. perda de pacotes (mestre)	Prob. perda de pacotes (escravo)	Pacotes enviados	Pacotes não enviados	Retransmissões
20%	20%	100	16	16
	30%	100	38	38
	40%	99	66	65
30%	20%	100	35	35
	30%	100	47	47
	40%	98	74	72
40%	20%	100	26	26
	30%	100	40	40
	40%	100	53	53

4.2 Simulação do DNP3 encapsulado em PLC

O DNP3 implementado teve suas funções básicas validadas na Seção 4.1. Nas simulações, foi feito o encapsulamento em TCP/IP, abstraindo as camadas inferiores, utilizando apenas um modelo probabilístico para os erros que podem ocorrer na camada de enlace de dados. As configurações realizadas nesta Seção desabilitam essa probabilidade de erros, atribuindo essa função ao PLC implementado.

O trabalho aqui proposto objetiva analisar o desempenho da rede, avaliando prioritariamente vazão e latência e comparando-as com os requisitos apresentados na Tabela 4.3. Além destes, são analisados outros parâmetros de qualidade, como perda de pacotes (número de retransmissões) e *jitter*.

Tabela 4.3 – Requerimento de Qualidade de Serviço (QoS) e tipos de tráfego em rede de comunicação inteligente.

Tipo de tráfego	Regularidade de tráfego	Taxa de dados	Latência
Comunicação interna na residência	Regular / sob demanda	Poucos kbps por dispositivo	2 ~ 15s
Leitura de medidor	Regular / sob demanda	Poucos bps ~ kbps por medidor	2 ~ 15s / centena de ms (para aplicações avançadas)
Conexão e desconexão	Ocasional	Taxa muito baixa	Longa (mudança de cliente) / centena de ms (resposta rápida para condições de rede)
Gerenciamento de interrupção	Ocasional	Taxa baixa	Próximo de tempo real (dezena de ms)
Resposta à demanda (DR)	Ocasional / sob demanda	Dezena de kbps	500ms (missão crítica) até alguns minutos para balanceamento de carga
Informação comercial de energia	Periódico	Taxa baixa	Dezena de segundos

Fonte: [Ho et al. \(2014\)](#) – Adaptada.

São propostas as seguintes configurações, com variação do esquema de modulação, da SNR e da taxa de ocupação do canal:

1. Avaliação de uma comunicação ponto a ponto para envio de dados DNP3;
2. Avaliação de uma configuração multiponto, com envio de dados DNP3.

4.2.1 Comunicação ponto a ponto para envio de dados DNP3

Uma configuração ponto a ponto consiste em um esquema de pequeno porte com ligação direta entre as estações, podendo ser dedicado, quando não há presença de outros dados na rede, ou não dedicado, quando há concorrência pelo canal. A configuração ponto a ponto simulada é dedicada e ilustrada na Figura 4.5, em que se tem uma estação mestre no poste da rede de distribuição e uma estação remota na residência. Sua topologia é mostrada na Figura 4.6.

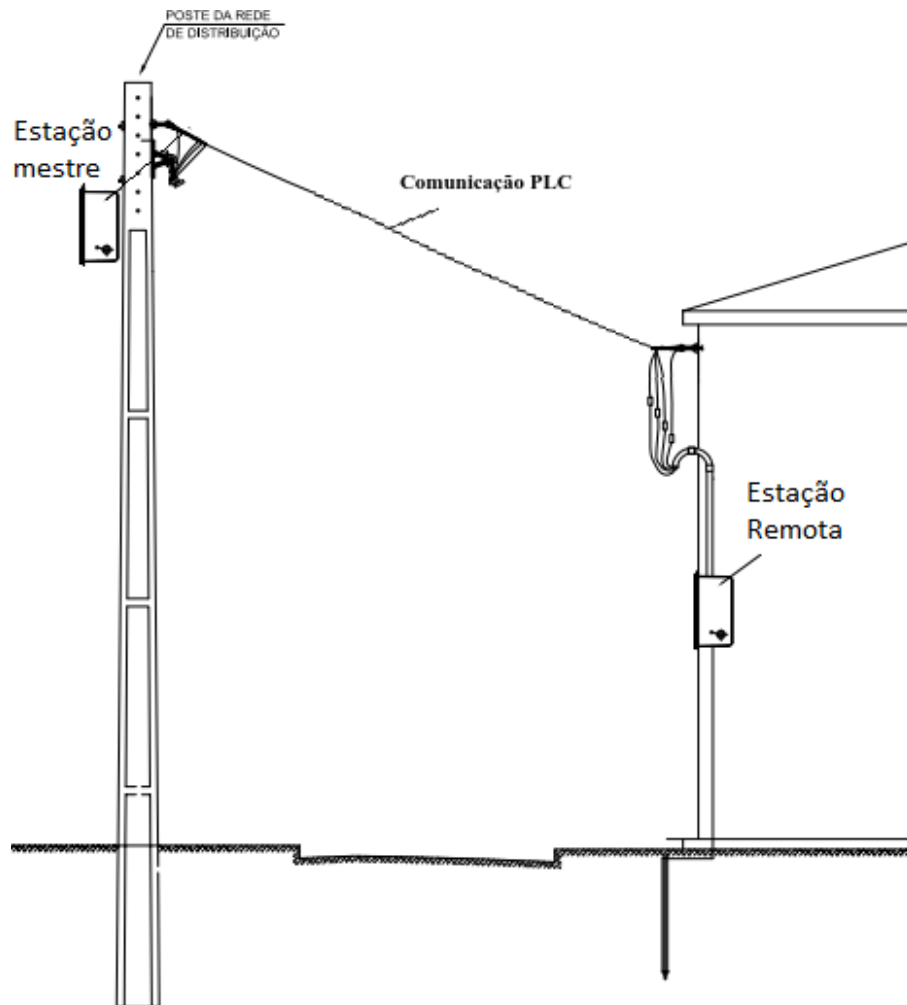


Figura 4.5 – Interligação da estação remota no consumidor à mestre na rede de distribuição.

Nesta configuração, foram avaliados os seguintes cenários:

1. Envio de uma mensagem não solicitada pela estação remota;
2. Solicitação de leitura de dados feita pelo mestre;
3. Envio de dados a serem escritos na estação remota;
4. Envio cíclico de mensagens não solicitadas.

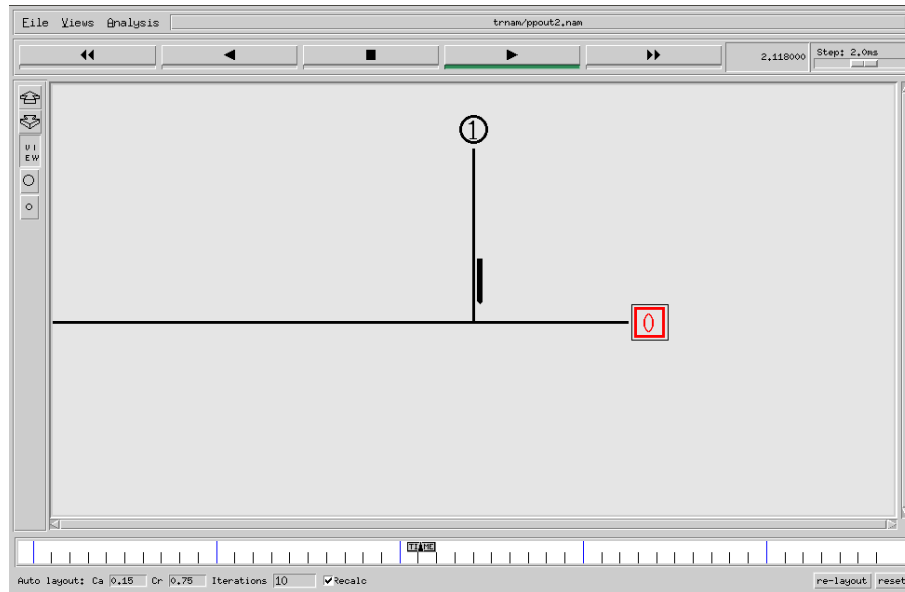


Figura 4.6 – Topologia do cenário de simulação da comunicação ponto a ponto.

4.2.1.1 Envio de uma mensagem não solicitada pela estação remota

A mensagem não solicitada pode ser enviada em intervalos periódicos (pré programados, como o envio constante de dados de medição), mas tendo como principal funcionalidade o envio de dados importantes, que precisam ser reportados à estação mestre.

No envio de apenas uma mensagem não solicitada à estação mestre, foi mantido um valor fixo de SNR (10 dB) e avaliados os níveis de vazão e latência para o envio de um pacote de 250 bytes, assim como o *jitter* para o envio dos pacotes que precisaram ser fragmentados, isto é, um pacote de 700 bytes e um pacote de 2000 bytes. A vazão obtida foi entre 36 – 38kbps para a modulação DQPSK e 18 – 19 kbps para DBPSK, sendo esses e os outros resultados apresentados na Tabela 4.4.

Tabela 4.4 – Envio de mensagens não solicitada na comunicação ponto a ponto.

Modulação	Distância entre estações (m)	Tamanho do pacote (bytes)	Vazão média (kbps)	Latência (ms)	<i>Jitter</i> (ms)
DQPSK	50	250	36,012	66,526	-
	1000	250	36,007	66,531	-
	1000	700	37,569	189,608	0,229
	1000	2000	38,433	533,851	0,229
DBPSK	50	250	18,191	131,350	-
	1000	250	18,190	131,355	-
	1000	700	18,834	378,084	2,477
	1000	2000	19,207	1068,177	2,477

Os dois primeiros envios realizados em cada esquema de modulação foram de um

pacote com 250 bytes, variando-se apenas a distância, utilizando 50m para a primeira simulação e 1km para a segunda. A variação dos resultados obtidos para latência e vazão se dá devido ao atraso de propagação no meio físico, explicado na Seção 3.2.1, correspondendo a uma diferença de 4,57bps na vazão média e de $5\mu s$ para a latência na modulação DQPSK e de 1,05bps na vazão média, com o mesmo valor para latência na modulação DBPSK.

Estes valores denotam que a influência da distância para o atraso de propagação não foi significativa. A distância influencia na intensidade da SNR do canal, tendo em vista que numa comunicação de dados a distância entre as estações provocam um aumento da atenuação e dos efeitos dos ruídos interferentes no canal.

Os valores de latência e vazão encontrados no envio dos pacotes de 250 bytes denotam um uso possível para aplicações em redes inteligentes, de acordo com as indicações da Tabela 4.3, exceto para gerenciamento de interrupção, que requer comunicação em tempo real, estando fora dos objetivos propostos.

Já os pacotes maiores têm uma aplicação mais restrita, inviabilizando o uso para os casos mais exigentes como missão crítica na resposta à demanda, caso o volume de dados seja em torno de 2kB, pois mesmo o DQPSK apresentou uma latência média maior que o limite de 500ms. O *jitter* desses pacotes foi pequeno comparado à latência, sendo de 0,229ms para DQPSK e 2,477ms para DBPSK. O *jitter* neste caso não constituiu um problema, pois a diferença nos atrasos foi apenas entre o primeiro e o segundo quadro.

4.2.1.2 Solicitação de leitura de dados feita pelo mestre

Numa comunicação entre as estações, um dos objetivos da estação mestre é a coleta periódica de dados do medidor de energia do consumidor, que pode ser para contabilizar seu faturamento mensal com leitura do consumo em kWh, ou para o caso de clientes que participam da geração distribuída, para levantamento da diferença entre o valor consumido e fornecido à rede elétrica.

Pode ser necessário também que a leitura do consumo seja feita não apenas do valor acumulado, como também do consumo horário, tendo em vista as mudanças na legislação que preveem a implantação de uma tarifa com horário de ponta, denominada de tarifa branca, em que o cliente é cobrado com valor diferenciado de acordo com o horário de consumo de energia².

Nesta simulação foi feita a solicitação de leitura de dados pelo mestre, enviando para isso pacotes com 200, 400 e 700 bytes, em ambas modulações utilizadas, encontrando os resultados apresentados na Tabela 4.5, em que se tem o tamanho do pacote de resposta enviado pela estação remota, sua carga de dados, bem como a latência e a vazão.

² A tarifa branca de energia foi aprovada pela ANEEL em março de 2016, por meio da sua resolução 733, estabelecendo as condições para aplicação, determinando sua implantação a partir de 1º de janeiro de 2018, para determinado grupo de consumidores, seguindo um cronograma para implantação completa.

Tabela 4.5 – Envio de mensagens de leitura na comunicação ponto a ponto.

Modulação	Tamanho do pacote enviado (bytes)	Tamanho do pacote de retorno (bytes)	Vazão média (kbps)	Latência (ms)
DQPSK	200	328	36,42	97,352
	400	695	37,89	188,914
	700	1156	38,18	314,042
DBPSK	200	328	18,17	195,15
	400	695	18,90	378,701
	700	1156	19,05	629,542

Os resultados apresentados são similares aos do envio de mensagem não solicitada, porém, neste caso, o tempo de retransmissão teve que ser ajustado. Este ajuste ocorreu pois o tamanho dos pacotes gerados pela estação remota foi maior que a confirmação gerada na mensagem não solicitada, e na solicitação de leitura feita pela estação mestre ocorreu troca de dados, como foi ilustrado na Figura 2.20.

4.2.1.3 Envio de dados a serem escritos na estação remota

O envio de escrita de dados na estação remota pode ocorrer para o caso de resposta à demanda, em que o cliente passaria a ter informação da rede para ajuste da sua carga. Isto é feito para balanceamento de cargas. No entanto uma alteração dinâmica da carga necessitaria de uma comunicação em tempo real, o que não está contemplado no presente trabalho.

Uma escrita de dados pode ocorrer também para mudança de configuração do medidor, como a alteração do intervalo de integração para o envio de mensagens não solicitadas. Na simulação foi realizada escrita de pacotes com 250, 700 e 2000 bytes, com seus resultados de latência e vazão média apresentados na Tabela 4.6.

Tabela 4.6 – Envio de mensagens de escrita na comunicação ponto a ponto.

Modulação	Tamanho do pacote enviado (bytes)	Vazão média (kbps)	Latência (ms)
DQPSK	250	37,05	66,531
	700	37,98	189,608
	2000	38,58	533,851
DBPSK	250	18,71	131,355
	700	19,04	378,084
	2000	19,28	1068,177

A escrita de dados difere também no envio do pacote de confirmação, que além de conter os 40 bytes do TCP/IP, conta com 17 bytes do DNP3, contendo 4 bytes de controle de aplicação, 1 byte de cabeçalho da função de transporte, 10 bytes de cabeçalho

de enlace e 2 bytes de CRC. A solicitação de leitura e escrita dos dados na estação remota não apresentaram valores significativos para o *jitter*.

4.2.1.4 Envio cíclico de mensagens não solicitadas

No primeiro envio cíclico de mensagens, foram dadas as mesmas condições para os dois esquemas de modulação, com uma utilização do canal inferior a 70%, para um tráfego livre de dados na rede.

Para SNRs mais baixas, a probabilidade de erro de bit torna-se maior, culminando em um aumento na perda de pacotes e necessidade de retransmissões, como pode ser observado no gráfico ilustrado na Figura 4.7. O DBPSK, por ser mais robusto, apresenta uma menor perda de pacotes que o DQPSK para uma mesma SNR, podendo ser uma opção mais viável quando o ambiente de rede for mais hostil e houver uma busca por maior confiabilidade da entrega da mensagem, e um menor requisito de alta vazão e baixa latência.

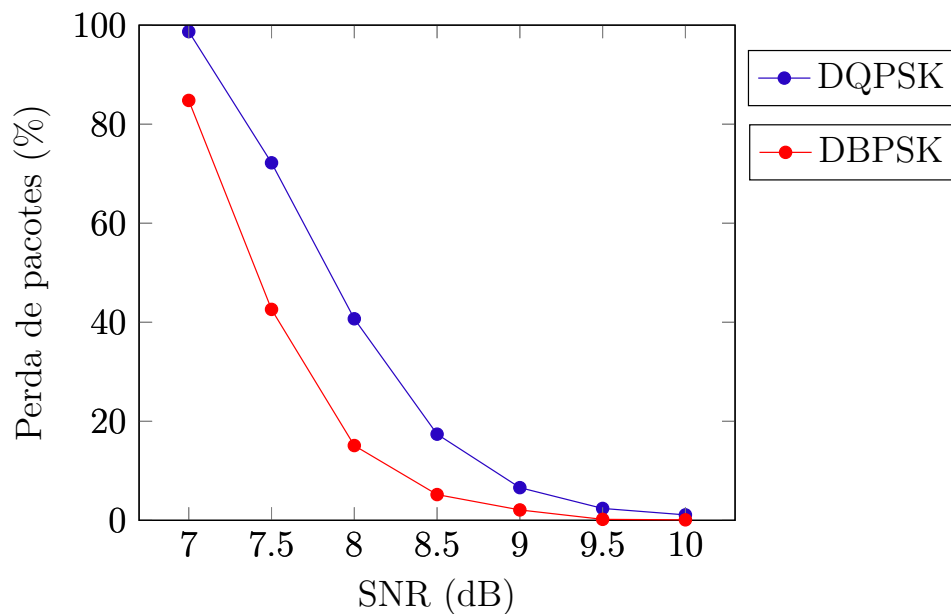


Figura 4.7 – Taxa de Perdas x SNR – Modulação DQPSK e DBPSK.

A latência média apresentada na simulação foi a mesma indicada na Tabela 4.4 para o envio de pacotes com 250 bytes, isto é, de 66,53ms para o DQPSK e 131,35ms para o DBPSK, havendo variação na vazão média de acordo com a SNR utilizada, como pode ser verificado no gráfico ilustrado na Figura 4.8.

Pode-se observar ainda que, para SNRs abaixo de 7,5dB, a vazão média de ambas as modulações diminuiu de forma expressiva. Contudo, neste caso, a modulação DBPSK apresentou valores maiores que a DQPSK devido ao aumento da perda de pacotes ser mais expressivo nessa última.

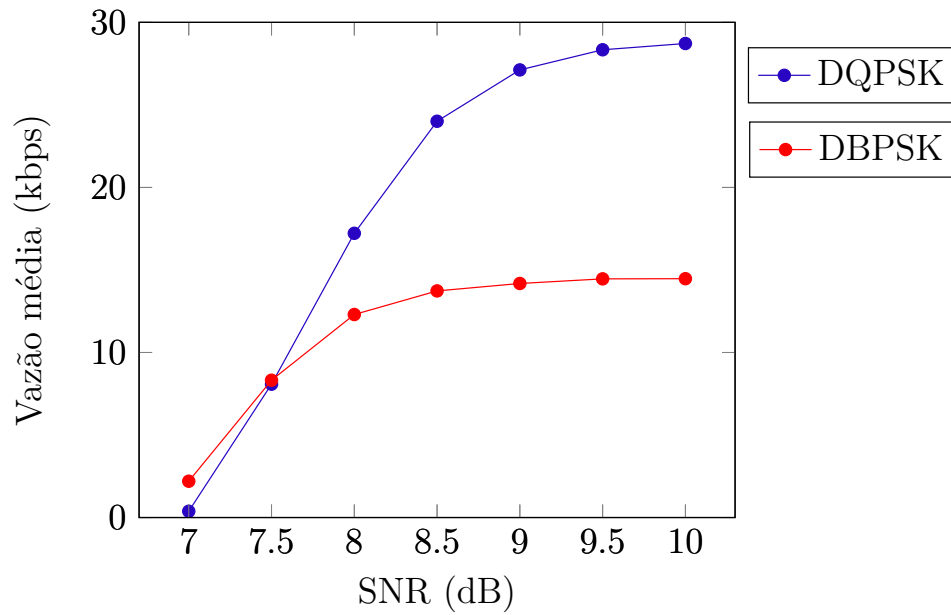


Figura 4.8 – Vazão média x SNR – Modulação DQPSK e DBPSK.

Na simulação seguinte, foi realizado o envio cíclico de dados com aumento crescente da taxa de ocupação próxima de 100%, tendo como resultado uma taxa de transmissão e latência médias constantes para ocupação de até 85%. Ao exceder essa ocupação de forma constante, aumentou-se a perda de pacotes e número de retransmissões, comprometendo consideravelmente a vazão média e a latência, como ilustrado nos gráficos das Figuras 4.9, 4.10 e 4.11.

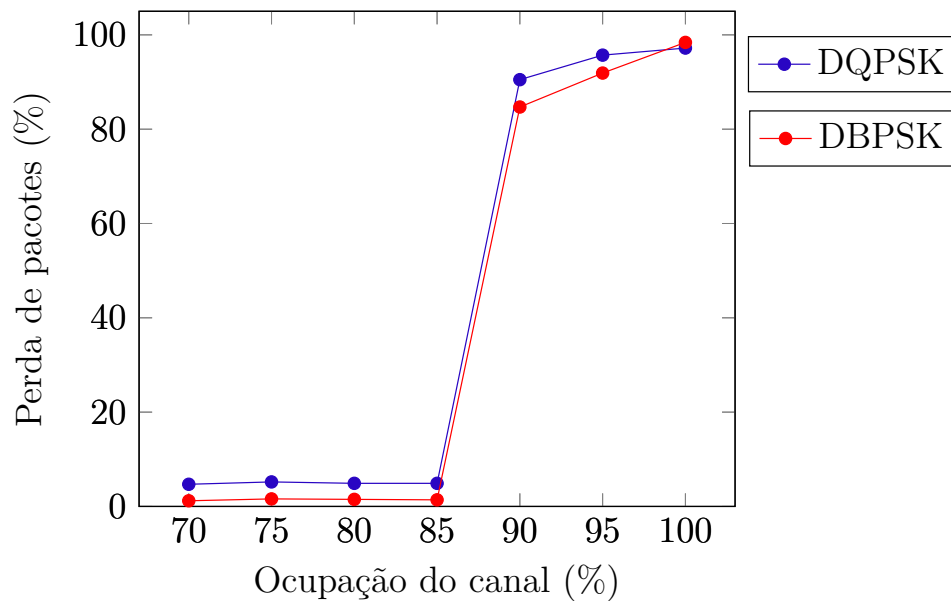


Figura 4.9 – Perda de Pacotes x Ocupação do canal – Modulação DQPSK e DBPSK.

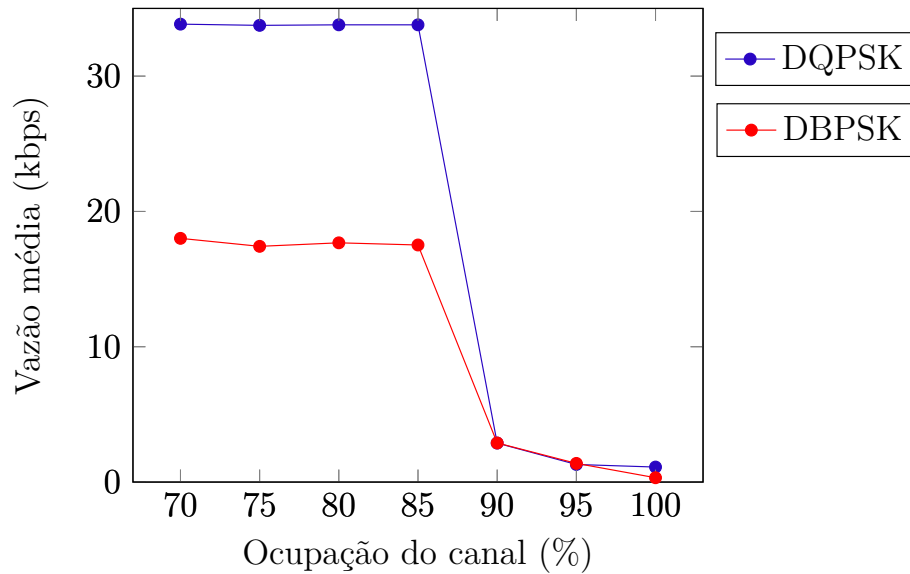


Figura 4.10 – Vazão média x Ocupação do canal – Modulação DQPSK e DBPSK.

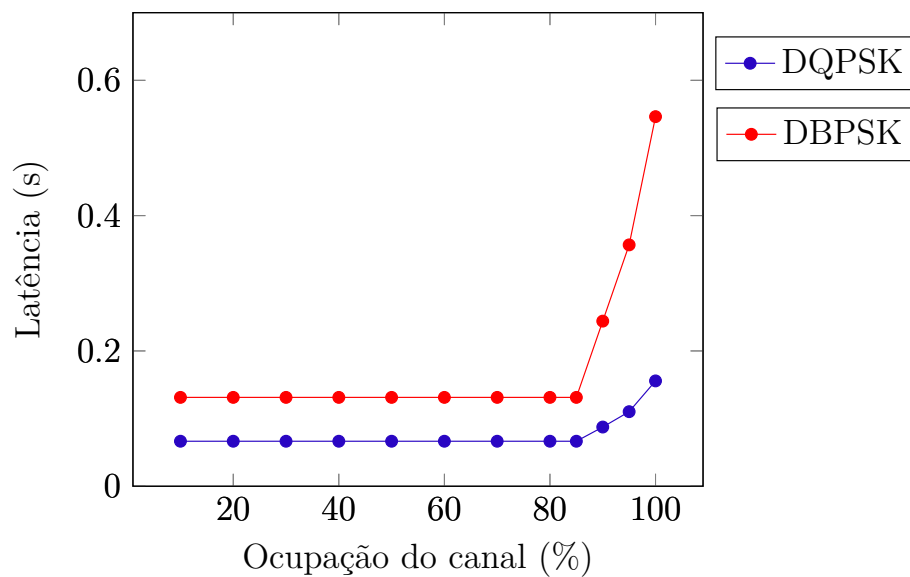


Figura 4.11 – Latência x Ocupação do canal – Modulação DQPSK e DBPSK.

O gráfico ilustrado na Figura 4.9 explica a mudança abrupta da vazão média e latência indicadas nos outros dois gráficos. Com o aumento da perda de pacotes, a comunicação tornou-se inexistente. Apesar de alguns pacotes terem sido entregues, a não confirmação desse processo à estação remota denota que a comunicação não foi realizada, necessitando assim o reenvio do pacote. Devido ao tempo de retransmissão ser fixo, com seu valor limite determinado pelo conteúdo do pacote (isto é, se o pacote é de leitura de um medidor de energia, ou ainda se é utilizado para missão crítica), a ocupação do canal extrapola o tempo de retransmissão, ocasionando um aumento da perda de pacotes.

Tabela 4.7 – Parâmetros iniciais da análise estatística da simulação.

Parâmetro	Valor
Número de simulações	10
Latência média (DQPSK)	65,531 ms
Latência média (DBPSK)	131,355 ms
Desvio padrão (DQPSK)	6,472
Desvio padrão (DBPSK)	12,139

4.2.1.5 Intervalo de confiança

O intervalo de confiança utilizado foi de 95% com um erro de 5%, valor conservativo apresentado na literatura e utilizado por [Jaimes \(2012\)](#). Para assegurar esse valor, foram realizadas inicialmente 10 simulações para com elas determinar uma média μ e um desvio padrão σ , cujos resultados são apresentados na Tabela 4.7.

A partir dos resultados, considerando a distribuição t de Student³, pôde-se determinar o número de simulações necessárias, por meio da Equação 4.1, em que:

$$Z_{a/2} \times \frac{\sigma}{\sqrt{n}} = e \quad (4.1)$$

- Z indica o coeficiente de confiança;
- a indica o nível de confiança;
- σ é o desvio padrão da amostra;
- n número de elementos da amostra;
- e representa o erro.

Com isso encontrou-se os valores apresentados na Tabela 4.8, indicando um mínimo de 15 simulações para cada cenário na modulação DQPSK e 14 para a DBPSK, variando para isso a *seed*, sendo tomado 15 como o número mínimo de simulações para cada esquema de modulação.

4.2.2 Comunicação multiponto para envio de dados DNP3

A comunicação multiponto difere da anterior pela presença de mais de uma estação remota ligada à mestre, como foi mostrado na Figura 2.21. Nas simulações, foram

³ A distribuição t de Student, é uma distribuição de probabilidade estatística utilizada quando o desvio padrão não é conhecido, utilizando para isso uma amostra como estimativa, normalmente inferior a 30.

Tabela 4.8 – Parâmetros iniciais da análise estatística da simulação.

Parâmetro	Valor
$Z_{\alpha/2}$	1,96
erro (DQPSK)	3,3266
erro (DBPSK)	6,5678
Número de simulações (DQPSK)	14,54
Número de simulações (DBPSK)	13,12

avaliados intervalos de integração diferentes para envio de mensagem não solicitada, sendo considerados dois cenários, um com o envio em rede dedicada, e outro numa rede com dados heterogêneos. Foi considerado também o envio das mensagens de leitura e escrita de dados para a estação remota.

Nas simulações foram utilizadas 10 estações remotas, numa configuração do tipo barramento, denotando um cenário típico de comunicação de uma estação mestre no poste da rede de distribuição, gerenciando uma rua, como é ilustrado na Figura 4.12. Esse cenário também é mostrado com uma animação do NS-2 apresentado na Figura 4.13.

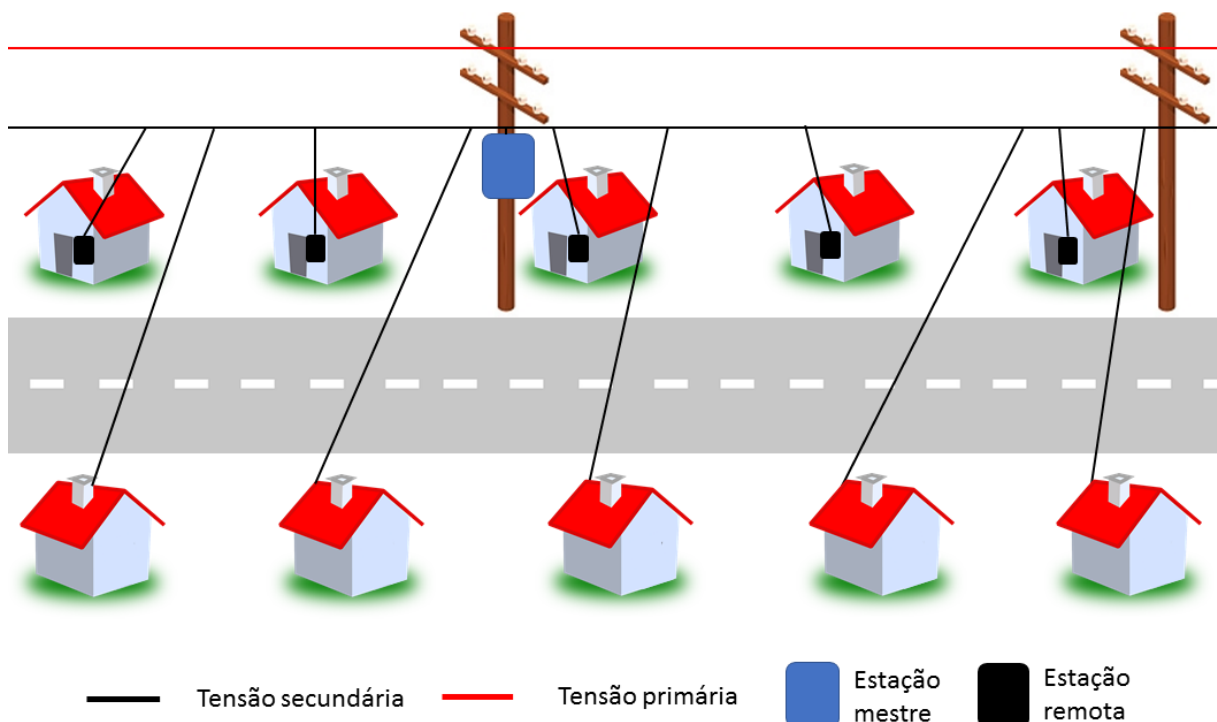


Figura 4.12 – Interligação das estações remotas no consumidor à mestre na rede de distribuição.

Para as simulações, foi editado o arquivo TCL, mudando os atributos de SNR e distância para cada estação, além da escolha do esquema de modulação.

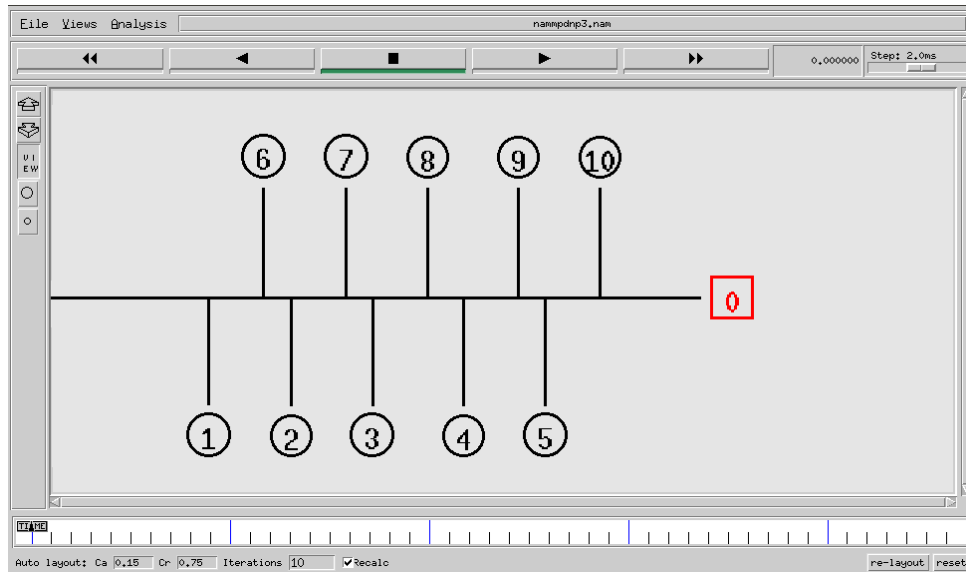


Figura 4.13 – Animação NAM da configuração multiponto.

4.2.2.1 Envio de mensagem não solicitada com rede dedicada

O envio constante de dados à estação mestre, é esperado em uma rede inteligente. Esses dados contém informações de consumo, fator de potência, além de parâmetros de qualidade da rede, como DHT – Distorção Harmônica Total. O envio desses dados deve ser realizado em um intervalo regular, denominado intervalo de integração, para que seja feito um acompanhamento dos dados.

O intervalo de envio dos dados utilizado por [Ortega \(2015\)](#) foi de 2 s, porém denota um intervalo com precisão excessiva, em que se tem um custo energético desnecessário para que sejam enviados dados, tendo em vista que uma das funções da estação remota é reportar qualquer alteração que ocorra, tornando-se desnecessário o envio exaustivo de mensagens à estação mestre.

Na escolha do intervalo de integração, foi levada em consideração a legislação vigente, que já é utilizada pelas concessionárias e permissionárias de energia elétrica para a coleta de dados dos consumidores do grupo A⁴, em que o intervalo utilizado é de 15 minutos para leitura de demanda.

Com isso foram propostos três intervalos de integração, 1 minuto, 5 minutos e 15 minutos, tendo em vista que atenderia grande parte das solicitações e, considerando um consumidor do grupo B, a máxima demanda instalada seria de 75kW. Para o pior caso, em que a unidade consumidora mantivesse essa demanda instalada por todo o tempo de fornecimento de energia, a variação máxima do consumo seria de 1,25kW por minuto, não sendo necessária a verificação em intervalos menores que esse, para leitura do consumo de

⁴ O grupo A se caracteriza pela utilização de tensão da rede primária, cujo esquema de tarifação possui uma tarifa binômia, faturando o consumo realizado pela unidade consumidora, com base no consumo e na demanda, cuja medição é realizada em intervalos de 15 minutos ([ANEEL, 2010b](#)).

energia.

Para realizar as simulações, teve-se como base a aplicação real, em que uma estação mestre poderia gerenciar 100 estações remotas. Sendo assim, o intervalo para envio de dados foi reduzido para 10% do seu valor, isto é, 6 s, para com isso reduzir o número de estações necessárias, sem perda de generalidade, tendo em vista o custo computacional necessário para realizar a simulação com um número grande de estações. Com isso, as simulações foram realizadas utilizando os parâmetros de entrada indicados na Tabela 4.9.

Tabela 4.9 – Parâmetros de entrada da simulação multiponto.

Unidade Consumidora	SNR (dB)	Distância (m)
1	8,5	300
2	8,9	200
3	9,0	100
4	8,8	400
5	8,7	500
6	8,5	600
7	8,4	500
8	8,2	500
9	8,0	600
10	7,8	700

Com os parâmetros indicados, foi realizado o envio de mensagens não solicitadas de 250 bytes, num total de 1440 mensagens por estação remota, correspondendo à simulação de um dia de envios de mensagem, em que foram avaliadas perdas de pacotes e latência de cada estação, bem como a vazão do canal.

Os resultados da simulação mostram a relação da perda de pacotes, com a SNR do canal, sendo ilustrada no gráfico da Figura 4.14. Pode-se ver um maior número de pacotes perdidos com a diminuição da SNR, bem como uma indicação menor de perdas para a modulação DBPSK.

A latência apresentada é similar às resultantes das simulações ponto a ponto, com alguns poucos pacotes apresentando valor maior que a média mostrada na Tabela 4.3, sendo maior esse quantitativo na modulação DQPSK que na DBPSK, como ilustrado nos histogramas apresentados nas Figuras 4.15 e 4.16.

A vazão de dados, ilustrada no gráfico da Figura 4.17, apresentou valores médios de 4,845kbps para a modulação DQPSK e 4,839kbps para DBPSK, respectivamente. A baixa vazão de dados dá-se devido ao quantitativo de pacotes enviados para o canal, tendo em vista não haver sobrecarga.

Considerando a latência apresentada, estima-se que o número de estações poderia ser aumentado, atendendo a um quantitativo estimado de 300 estações remotas, sem

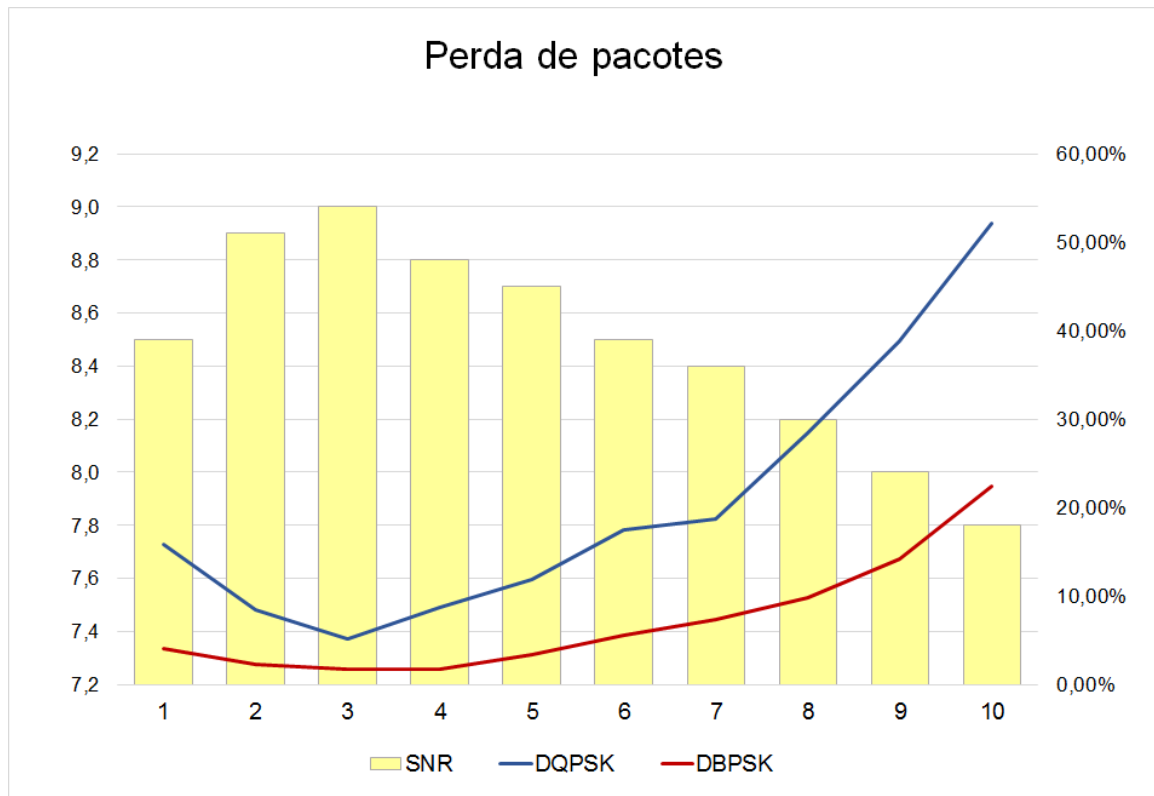


Figura 4.14 – Perda de pacotes x SNR.

sobrecarregar a rede de comunicação, com qualquer uma das duas modulações utilizadas, tendo em vista que a maior média de latência envolvida, do DBPSK, foi inferior a 150 ms.

Esse cenário típico, de um canal dedicado, é justificável em um condomínio, em que uma estação mestre gerencia dados das unidades consumidoras, funcionando como uma estação intermediária, isto é, reportando as informações a outra estação hierarquicamente superior.

Outro ponto a se destacar foi o aumento do intervalo de integração, que permite um gerenciamento de mais unidades consumidoras, tendo como contrapartida uma diminuição da quantidade de dados coletados de cada estação.

4.2.2.2 Envio de outras funções junto ao envio de mensagens não solicitadas

Uma situação mais realista, ainda utilizando uma rede dedicada, dá-se com o envio de solicitações do mestre, além das mensagens não solicitadas reportadas. Nesta simulação, foi acrescentada à anterior as funções de leitura e escrita de dados da estação remota. Para implementação foi alterado o código TCL, criando e utilizando as variáveis mostradas na Tabela 4.10.

O envio de mensagens não solicitadas em cada estação é feita em um laço de repetição, cujo tempo total é determinado pela quantidade de mensagens a ser enviada e o

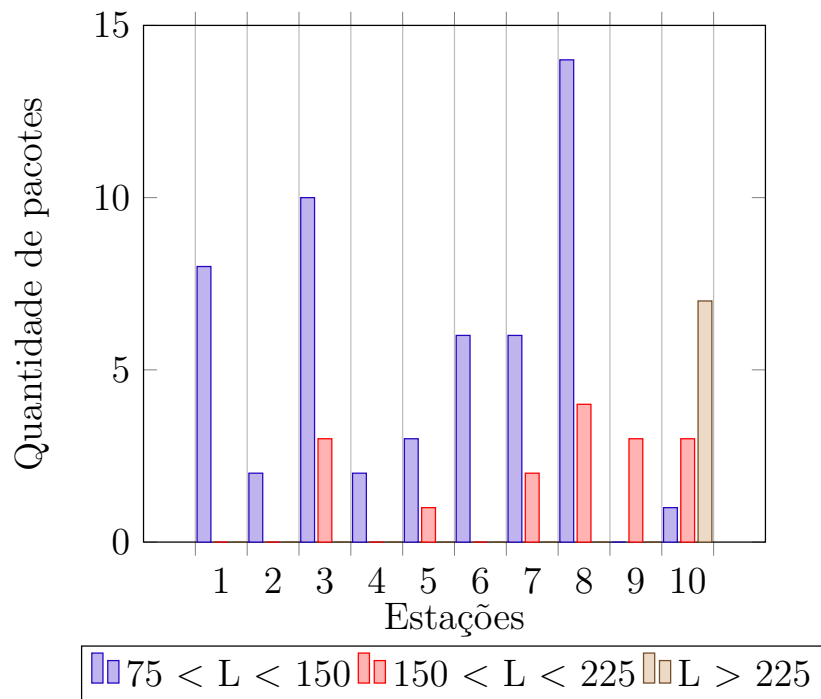


Figura 4.15 – Quantidade de pacotes com latência acima da média – DQPSK.

Tabela 4.10 – Parâmetros de entrada – outras funções.

Parâmetro	Variação	Descrição
result_envio	0 – 100	Probabilidade da estação mestre fazer uma solicitação, igual a 5%.
result_tipo	1 – 2	Tipo de dado que será enviado, isto é, se o dado será de leitura ou escrita da estação.
result_instante	0 – 6	Instante em que o dado será enviado, variando de 0 ao intervalo de integração.
result_tamanho	100 – 500	Tamanho do pacote enviado, com valores de 100 a 500 bytes.
result_estacao	1 – 10	Escolhe a estação para a qual será feita a solicitação.

intervalo de transmissão entre elas. Com isso, para o envio de outras mensagens de forma aleatória, foi introduzido dentro deste laço outro laço, com as atribuições para o envio aleatório de solicitação de leitura e escrita de dados.

O envio é realizado se o valor da variável `result_envio` for menor que 5, representando uma probabilidade de 5% de envio de outras mensagens, no intervalo de simulação. Sendo confirmado o envio, é então escolhido o tipo de mensagem, de leitura (1) ou escrita (2), atribuindo a variável “evento” a mensagem correspondente “*read*” ou “*write*”.

Em seguida é verificada a estação escolhida, atribuindo à variável “cliente” a estação correspondente (client1 para estação 1, client2 para a estação 2...) e por fim é dado o

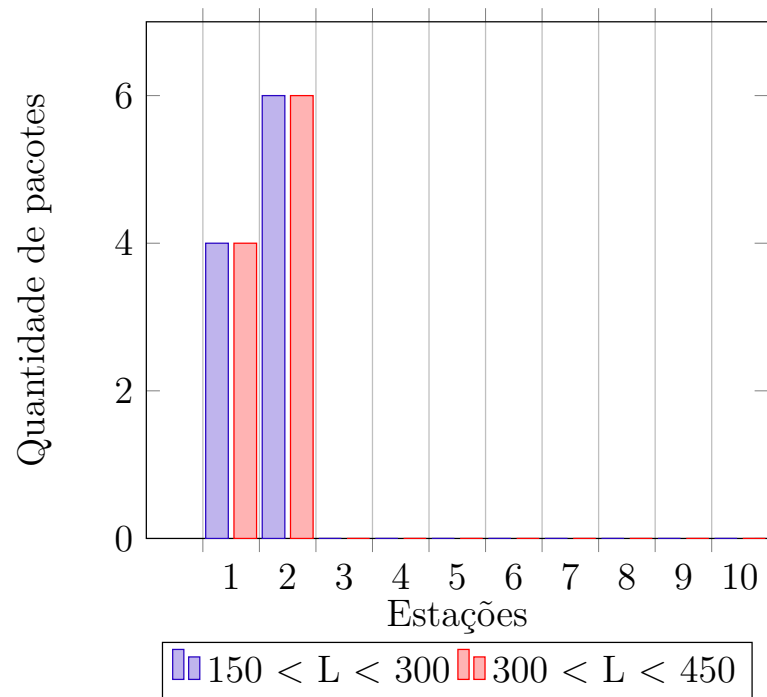


Figura 4.16 – Quantidade de pacotes com latência acima da média – DBPSK.

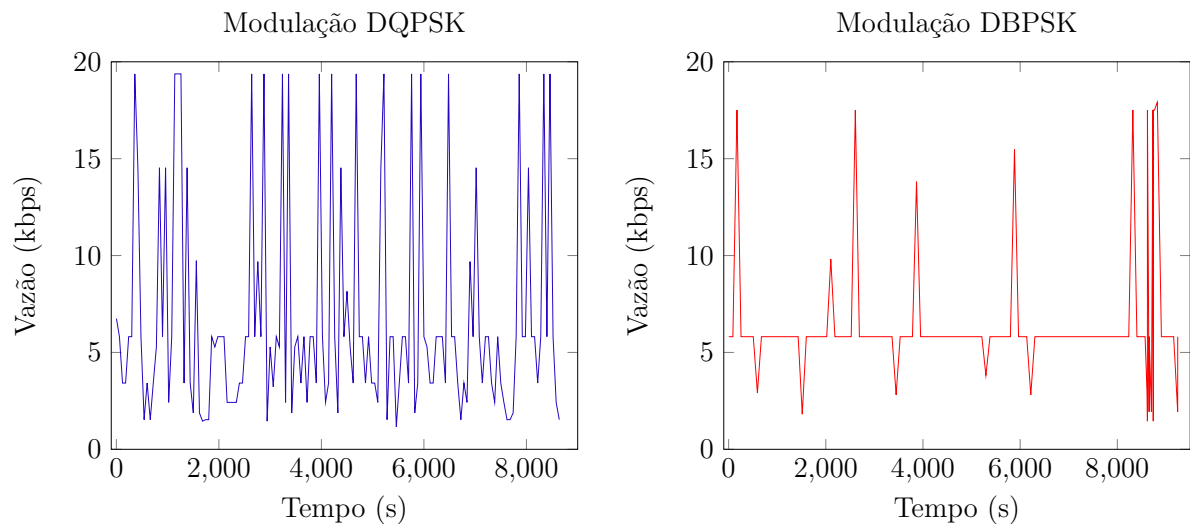


Figura 4.17 – Vazão de dados – Modulação DQPSK e DBPSK.

comando para que seja enviado o pacote, utilizando a seguinte expressão:

`$ns at [expr $i + $result_instante] "$master $evento $cliente $result_tamanho"`

em que se tem no instante resultante da expressão (expr), o envio do \$evento da estação mestre (master) para a estação \$cliente, contendo o tamanho gerado pela variável aleatória \$result_tamanho. Os resultados do envio de mensagens geradas são apresentados na Tabela 4.11.

Tabela 4.11 – Número de pacotes enviados na simulação multiponto – DQPSK.

Estação	DQPSK		DBPSK	
	Leitura	Escrita	Leitura	Escrita
1	4	2	2	0
2	6	4	6	4
3	2	0	8	8
4	2	6	2	0
5	8	0	0	0
6	2	8	2	4
7	0	6	4	6
8	0	2	6	2
9	0	2	4	4
10	0	4	2	0

O envio de outros dados, dentro do cenário de envio cíclico de dados não alterou os valores resultantes de latência média do canal e o *jitter* apresentado pelos pacotes fragmentados foram desprezíveis. Em relação a vazão média, não houve aumento significativo, devido ao fluxo de dados existente não comprometer a ocupação do canal, apresentando latências condizentes com as apresentadas na comunicação ponto a ponto. Apenas 4 pacotes, da modulação DQPSK, precisaram ser retransmitidos, o que é desprezível diante de um tráfego total de mais de 15000⁵ mensagens.

4.2.2.3 Envio de mensagem não solicitada com rede de dados heterogêneos

Em muitos casos, a rede não é dedicada, havendo o fluxo de outros tipos de dados de ou para a concessionária de energia. Na simulação proposta, o fluxo de dados utilizado foi o CBR, fazendo com que o mesmo fizesse o envio de dados constantes, porém em intervalos variáveis para alterar a taxa de ocupação do canal, verificando com isso a resposta da comunicação DNP3 a essa variação. A animação NAM do cenário simulado é ilustrada na Figura 4.18, em que os nós 11 e 12 são responsáveis pelo envio e recebimento do tráfego de dados, respectivamente.

A presença de dados heterogêneos tem por consequência a variação na ocupação do canal. Isso proporciona períodos com aumentos ou diminuições dos parâmetros de qualidade avaliados. A ocupação do canal no decorrer do tempo simulado é ilustrada nos gráficos da Figura 4.19.

Pode-se ver que, no período simulado, o canal apresentou ocupações de até 90%, que para os dois esquemas de modulação empregados, alguns pacotes apresentaram um aumento expressivo na latência com o aumento da ocupação do canal, como é ilustrado nos gráficos da Figura 4.20, em que se tem variação da latência do canal de acordo com a taxa de ocupação.

⁵ As 14400 mensagens não solicitadas acrescidas das retransmissões.

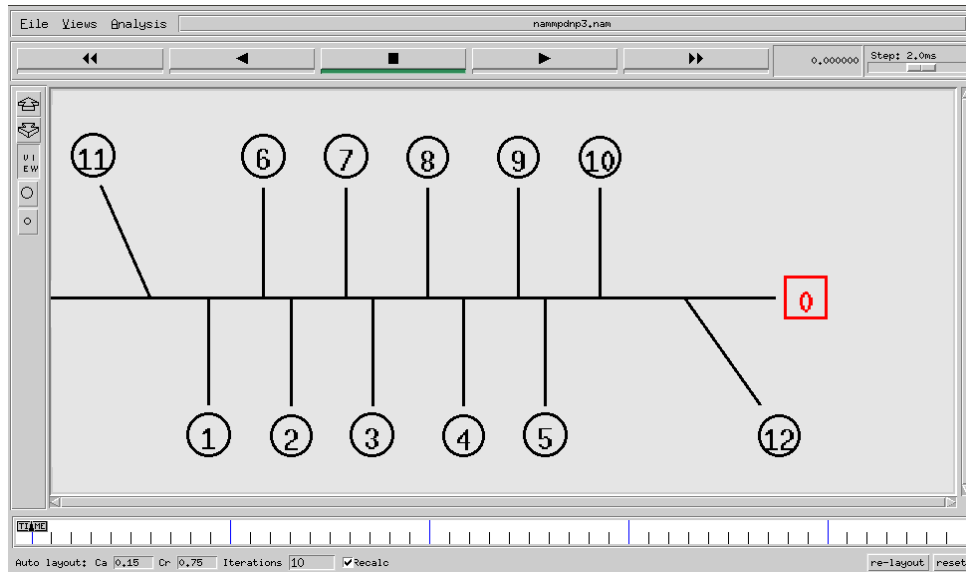


Figura 4.18 – Animação NAM da configuração multiponto com tráfego heterogêneo.

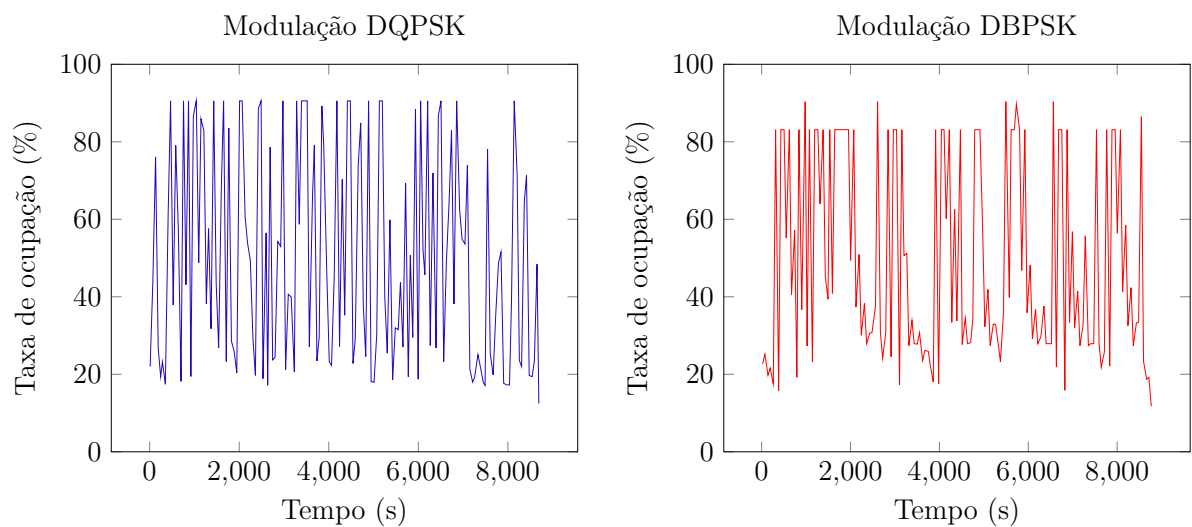


Figura 4.19 – Ocupação do canal na rede com tráfego heterogêneo.

O aumento da ocupação do canal indicou também o aumento da latência dos dados e que, apesar de a Tabela 4.4 mostrar o DBPSK com o dobro da latência média do DQPSK, este apresentou uma quantidade menor de pacotes com aumento da latência, não indicando nenhum pacote com valores superiores a 500 ms, para ocupação do canal em até 85%. Com a ocupação em torno de 90%, o DBPSK também apresentou uma latência menor, de 1225,300 ms contra 1701,291 ms apresentado pelo DQPSK.

Pode-se ver ainda, por meio da Tabela 4.12 que a robustez da modulação DBPSK culminou numa latência média próxima a apresentada na DQPSK, que para a modulação ponto a ponto indicou valores duas vezes maiores, ou seja, cerca de 65,53 ms no DQPSK e 131,35 ms para o DBPSK. Com a geração de dados aleatória, a latência máxima é

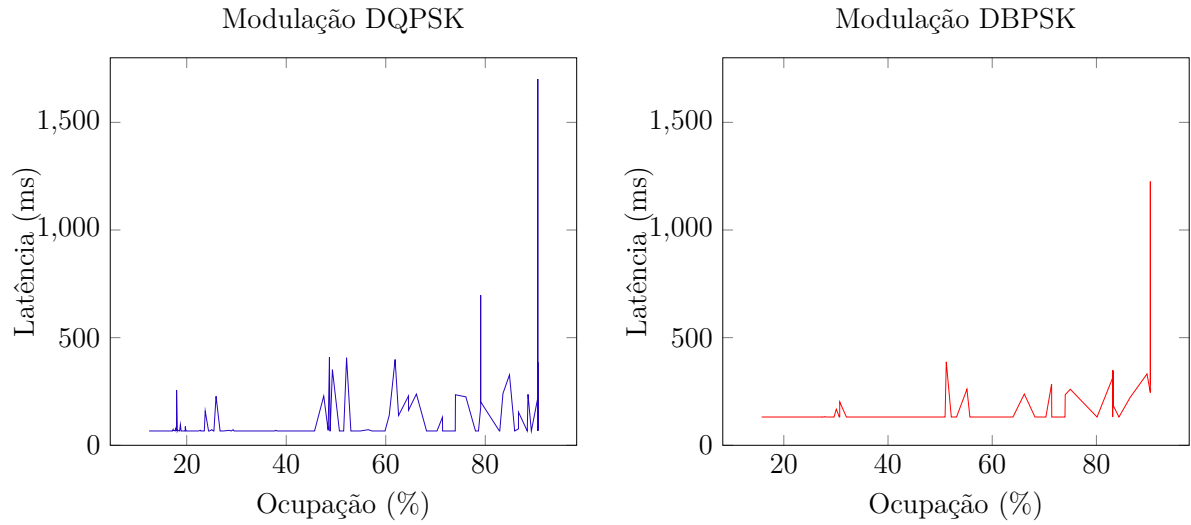


Figura 4.20 – Latência de dados x Taxa de Ocupação – rede de tráfego heterogêneo.

verificada em instantes diferentes, ilustrados nos gráficos da Figura 4.21.

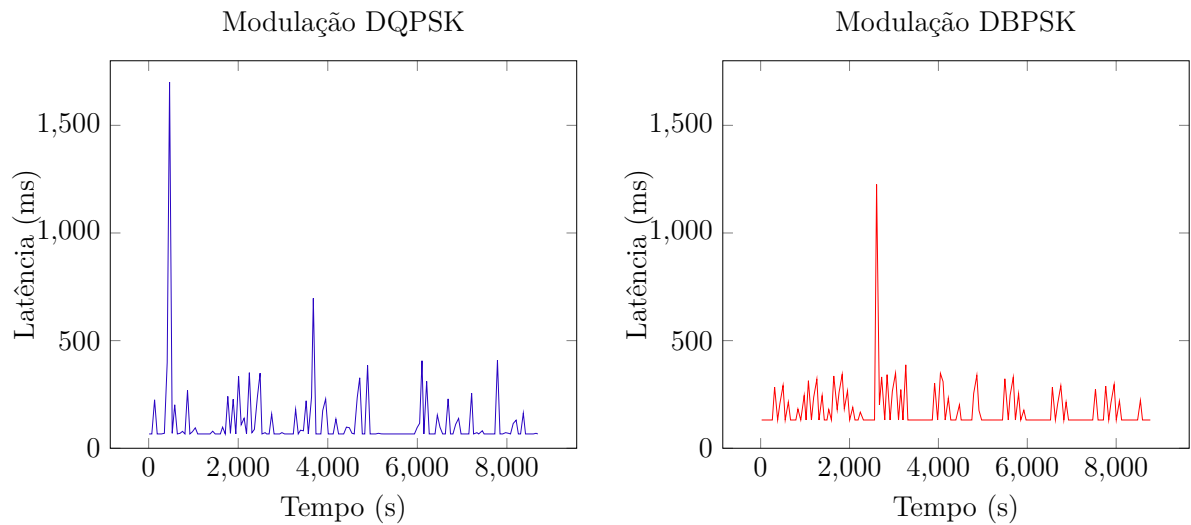


Figura 4.21 – Latência de dados em rede com tráfego heterogêneo.

A vazão também indicou variações com o tempo, devido à variação do fluxo de dados, como é ilustrado nos gráficos da Figura 4.22. A vazão média apresentada foi de 4,789 kbps para a modulação DQPSK e de 4,839 kbps para a modulação DBPSK, sendo esses e outros resultados da simulação de rede com tráfego heterogêneo mostrados na Tabela 4.12, da qual, pode-se ver taxas de ocupação do canal similares para ambas as modulações, com valores médios pouco maiores que 40%.

As simulações demonstraram viabilidade no uso do DNP3 sobre PLC, apresentando bons resultados de latência, tanto numa rede de dados com tráfego homogêneo, quanto heterogêneo para, a maioria das aplicações em redes inteligentes indicadas na Tabela

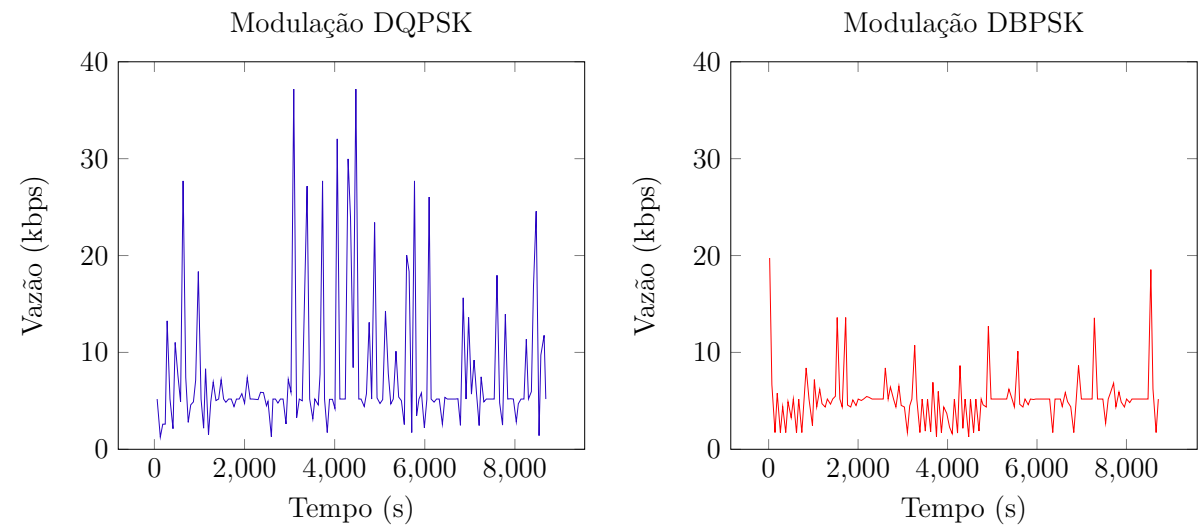


Figura 4.22 – Vazão de dados em rede com tráfego heterogêneo.

Tabela 4.12 – Envio de pacotes em rede com tráfego heterogêneo.

Modulação	Latência (ms)			Vazão(kbps)			Pacotes perdidos		Ocupação(%)		
	Min	Med	Max	Min	Med	Max	Qtd	%	Min	Med	Max
DQPSK	65,527	119,118	1701,291	0,609	4,789	39,126	469	32,57	4,63	43,09	91,20
DBPSK	131,35	152,989	1225,299	1,282	4,839	19,453	97	6,74	4,48	41,65	90,35

4.3, sendo preferível a modulação DBPSK quando num ambiente bastante ruidoso, por apresentar uma menor perda de pacotes.

Os cenários simulados apresentaram bons resultados, com uso indicado para leitura do medidor, conexão e desconexão de redes para mudança de cliente, informação comercial de energia e resposta à demanda para balanceamento de cargas.

5 Conclusões

A utilização de uma rede de comunicações de forma harmoniosa com uma rede de energia elétrica é um dos principais desafios na implantação de uma das principais características de uma rede inteligente. Essa comunicação dá-se por meio de diversas tecnologias, de forma a transpor as barreiras físicas e muitas vezes geográficas que existem em uma comunicação de dados.

A escolha da tecnologia a ser utilizada dá-se de forma a atender aos critérios técnicos necessários, como confiabilidade e velocidade de transmissão, sendo avaliado também o custo necessário para sua implantação. O PLC destaca-se por ser uma tecnologia economicamente viável, quando comparado às alternativas, como *Ethernet* ou *ZigBee*, o que se dá por não necessitar da instalação de novos condutores, utilizando a rede elétrica existente.

A rede elétrica foi desenvolvida apenas para a condução de eletricidade, proporcionando ruídos e atenuações na comunicação PLC, que resultam numa baixa SNR, comprometendo de forma significativa a transmissão de dados. Esse comprometimento dá-se com um aumento da taxa de erro de bits e, por consequência, da taxa de erro de pacotes. Com uma grande quantidade de pacotes perdidos, são necessárias retransmissões, aumentando a latência e diminuindo a vazão de dados.

Este trabalho contribuiu com a modelagem da comunicação PLC no simulador de redes NS-2, permitindo a simulação de cenários de comunicação utilizando o protocolo DNP3, que já é empregado em comunicação de subestações, de forma a estabelecer uma base comparativa de desempenho, que pode referenciar implementações ou adaptações de redes futuras.

Os resultados demonstraram uma viabilidade no uso do DNP3 encapsulado em PLC para aplicações em redes inteligentes, tanto para modulação DQPSK quanto para DBPSK, excetuando-se as atividades que necessitam de comunicação em tempo real, como é o caso do gerenciamento de conexões, ou ainda atividades de missão crítica, existente na resposta à demanda.

Dentre as possibilidades de uso estão a leitura dos medidores de energia, resposta à demanda para balanceamento da carga, informação comercial de energia, assim como mudança de cliente, o que já permite uma maior transparência no fornecimento de energia ao consumidor, tendo em vista que ele pode, além de acompanhar seu consumo de energia de forma mais dinâmica e interativa, estar atualizado a respeito da qualidade da energia que está recebendo da concessionária de modo a interagir com o sistema, como é o caso do balanceamento de cargas.

Comparando com os resultados de uma rede ZigBee, as simulações de [Pereira \(2015\)](#) apresentaram bons resultados quanto à latência para redes com uma pequena quantidade de nós, indicando seu uso para comunicação em tempo real, mas que pode ser comprometido com a distância, necessitando de um maior número de saltos, o que além de encarecer o seu uso, irá aumentar a latência a níveis não permitidos para a comunicação em tempo real.

Como continuidade deste trabalho sugere-se melhorar a modelagem do canal PLC, utilizando outras variáveis de entrada como distância e impedância das cargas, para com elas obter o valor da SNR e com isso obter uma melhor precisão na implementação. Além disso, pode-se verificar a atuação do DNP3 em cenários com estações concentradoras de dados e múltiplos mestres, agregando a tecnologia PLC com o ZigBee e a *Ethernet*, verificando inclusive a sua utilização como redundância no sistema.

Referências

- ANATEL. Resolução normativa nº 527, de 08 de abril de 2009. aprova o regulamento sobre condições de uso de radiofrequências por sistemas de banda larga por meio de redes de energia elétrica. **Diário Oficial da República Federativa do Brasil**, p. 9, Abr. 2009. Disponível em: <<http://www.anatel.gov.br/legislacao/resolucoes/2009/101-resolucao-527>>. Acesso em: 15 set 2016. Citado 2 vezes nas páginas 22 e 23.
- ANEEL. Chamada nº 011/2010 projeto estratégico”: Programa brasileiro de rede elétrica inteligente. **Agência Nacional de Energia Elétrica**, Jul. 2010. Disponível em: <http://www.aneel.gov.br/arquivos/PDF/PeD_2008-ChamadaPE11-2010.pdf>. Acesso em: 29 jun 2015. Citado na página 2.
- _____. Resolução normativa nº 414, de 09 de setembro de 2010. Estabelece as condições gerais de fornecimento de energia elétrica de forma atualizada e consolidada. **Diário Oficial da República Federativa do Brasil**, p. 196, Set. 2010. Disponível em: <<http://www.aneel.gov.br/cedoc/ren2010414.pdf>>. Acesso em: 15 jun 2015. Citado na página 70.
- _____. Resolução normativa nº 687, de 24 de novembro de 2015. altera a resolução normativa nº 482, de 17 de abril de 2012, e os módulos 1 e 3 dos procedimentos de distribuição – prodist. **Diário Oficial da República Federativa do Brasil**, p. 25, Nov. 2015. Disponível em: <<http://www2.aneel.gov.br/cedoc/ren2015687.pdf>>. Acesso em: 28 set 2016. Citado na página 1.
- _____. Ficheiro:matrizenergetica.jpg. **ANEEL**, ANEEL, 2016. Disponível em: <<http://www.aneel.gov.br/aplicacoes/capacidadebrasil/OperacaoCapacidadeBrasil.cfm>>. Acesso em: 09 nov 2016. Citado na página 13.
- AUGUSTO, Ricardo de Sousa. Simulation of powerline communication (plc) for smart grids in omnet+. 2012. Citado 5 vezes nas páginas 5, 47, 48, 49 e 50.
- BANI-AHMED, Abedalsalam et al. Microgrid communications: State of the art and future trends. p. 780–785, 2014. Citado na página 7.
- BONGIOLO, Guilherme Grazziotin; MIRANDA, Tiago André de. Smart grid e plc: uma análise da regulação europeia e desenvolvimento de um sistema simples de integração destas tecnologias. 2014. Citado na página 6.
- BUSH, Stephen F. **Smart Grid: Communication-Enabled Intelligence for the Electric Power Grid**. New Delhi, India: John wiley & sons, 2014. Citado na página 3.
- CHAMPATI, Nayak Ajit Kumar Rai Satyananda; RAJIB, Mall. **Computer Network Simulation Using NS2**. New York, USA: CRC Press, 2016. Citado 2 vezes nas páginas 14 e 15.
- CLARKE, G; REYNOLDS, D. **Practical Modern SCADA Protocols: DNP3, IEC 60870.5 and Related Systems**, Newnes. United Kingdom: Oxford, 2004. Citado 5 vezes nas páginas 2, 3, 33, 34 e 35.

COELHO, Francisco Jonatas Siqueira. **Uma infraestrutura para o monitoramento do consumo de energia residencial baseado em Power Line Communication**. 107 p. Dissertação (Mestrado) — Universidade Federal de Pernambuco – UFPE, 2015. Citado 5 vezes nas páginas 2, 6, 19, 20 e 21.

COUTINHO, Mauro Margallo et al. Avaliando o desempenho de redes através do simulador network simulator. **Belém: Universidade Federal do Pará**, 2003. Citado na página 14.

DARWISH, Ihab et al. Smart grid dnp3 vulnerability analysis and experimentation. p. 141–147, 2015. Citado na página 8.

DEPURU, Soma Shekara Sreenadh Reddy; WANG, Lingfeng; DEVABHAKTUNI, Vijay. Smart meters for power grid: Challenges, issues, advantages and status. **Renewable and Sustainable Energy Reviews**, Elsevier, USA, n. 2736–2742, Fev. 2011. Citado na página 12.

DIAS, Paulo Rogério et al. Modelagem da rede de distribuição de energia elétrica para aplicações de canais plc (power line communications). **Congresso Internacional de Distribuição Elétrica – CIDEL**, Argentina, 2010. Citado na página 4.

EMPRESA DE PESQUISA ENERGETICA. Projeto da usina hidrelétrica de belo monte: Fatos e dados. **Ministério de Minas e Energias**, Fev. 2011. Disponível em: <http://www.mme.gov.br/documents/10584/1590364/BELO_MONTE_-_Fatos_e_Dados.pdf/94303fc2-d171-45be-a2d3-1029d7ae5aad>. Acesso em: 20 jul 2015. Citado na página 12.

FADAEENEJAD, M. et al. The present and future of smart power grid in developing countries. **Renewable and Sustainable Energy Reviews**, Elsevier, USA, n. 828–834, Ago. 2013. Citado na página 13.

FANG, Xi et al. Smart grid – The New and Improved Power Grid: A Survey. **IEEE Communications Surveys & Tutorials**, IEEE, S.l., n. CT-18, p. 944 – 980, out. 2012. Disponível em: <<http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=9739>>. Acesso em: 03 mai 2015. Citado na página 10.

FERREIRA, Hendrik C; LAMPE, Lutz. **John Newbury, Theo G. Swart. Power Line Communications: Theory and Applications for Narrowband and Broadband Communications over Power Lines**. Torquay, UK: Wiley, 2010. Citado 2 vezes nas páginas 3 e 18.

FOROUZAN, Behrouz A. **Comunicação de dados e redes de computadores**. Brasil: AMGH Editora, 2009. Citado 3 vezes nas páginas 24, 30 e 51.

FRANÇA, André Morelato et al. A tecnologia plc: Oportunidade para os setores de telecomunicações e energia elétrica. **TELECO**, TELECO, 2006. Disponível em: <<http://www.teleco.com.br/tutoriais/tutorialkbns/default.asp>>. Acesso em: 10 jan 2016. Citado na página 22.

GALLI, Stefano; SCAGLIONE, Anna; WANG, Zhifang. Power line communications and the smart grid. p. 303–308, 2010. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5622060>>. Citado 3 vezes nas páginas 20, 24 e 25.

GOODE, Bur. Voice over internet protocol (voip). **Proceedings of the IEEE**, IEEE, v. 90, n. 9, p. 1495–1517, 2002. Citado na página 20.

HO, Quang-Dung et al. Smart grid communications network (sgcn). In: **Wireless Communications Networks for the Smart Grid**. New York, USA: Springer, 2014. p. 15–30. Citado na página 60.

IEEE. Ieee standard for electric power systems communications–distributed network protocol (dnp3). **IEEE Standard**, IEEE, 2012. Disponível em: <<https://standards.ieee.org/findstds/standard/1815-2012.html>>. Acesso em: 12 dez 2016. Citado 9 vezes nas páginas 3, 26, 27, 28, 29, 30, 34, 36 e 37.

_____. Ieee standard for low-frequency (less than 500khz) narrowband power line communications for smart grid applications. **IEEE Standard**, IEEE, 2013. Disponível em: <<https://standards.ieee.org/findstds/standard/1901.2-2013.html>>. Acesso em: 15 jan 2017. Citado na página 51.

INSTITUTE, Information Sciences. Nam. **USCVirtebi**, School of Engineering, 2016. Disponível em: <<http://www.isi.edu/nsnam/nam/>>. Acesso em: 13 ago 2016. Citado 2 vezes nas páginas 4 e 37.

ISSARIYAKUL, Teerawat; HOSSAIN, Ekram. **Introduction to network simulator NS2**. Florida, USA: Springer Science & Business Media, 2011. Citado 3 vezes nas páginas 15, 16 e 42.

JAIMES, Oscar Eduardo Ramírez. **Estudios de desempeño de escenarios SCADA que utilizan el Protocolo DNP3**. Dissertação (Mestrado) — Uniandes, 2012. Citado 11 vezes nas páginas 3, 4, 31, 32, 37, 38, 39, 40, 44, 56 e 68.

JIN, Cheng; KUNZ, Thomas. **Design and Implementation of a Smart Home Networking Simulation**. Ottawa, CA, 2010. Citado na página 51.

KELLERBAUER, Holger; HIRSCH, Holger. Simulation of powerline communication with omnet++ and inet-framework. p. 213–217, 2011. Citado na página 5.

KUMAR, Prashant; MAHESHWARI, Shivangi; DUBEY, Harsh Kumar. Development and validation of a ns2 protocol for data security at network layer. p. 529–534, 2015. Citado na página 8.

KUROSE, James F.; ROSS, Keith W. **Redes de computadores e a Internet: uma abordagem top–down**. São Paulo, SP: Pearson Education do Brasil Ltda, 2013. Citado 2 vezes nas páginas 3 e 50.

LEITE, Davi Rabelo Viana. **Medidores eletrônicos: Análise de viabilidade econômica no contexto das redes inteligentes**. 81 p. Dissertação (Mestrado em Engenharia Elétrica) — Universidade de Brasília, Brasília /DF/BR, 2013. Citado na página 10.

LOBATO, FGideon Igor Carvalho. **Sistema de Comunicação e Comando Remoto de Inversores Utilizando o Barramento de Alimentação em Corrente Contínua**. 97 p. Dissertação (Mestrado) — Universidade Federal de Minas Gerais – UFMG, 2014. Citado na página 6.

LU, Xiaolin; MONNIER, Olivier; DALLAS, PLC. Developing robust power line communications (plc) with g3. **Texas Instruments**, 2012. Citado na página 18.

- MATHIAS, Luis Carlos. **Análise, Modelagem e Implementação de um Sistema de Comunicação via Rede Elétrica para Domótica**. Dissertação (Mestrado) — Universidade Estadual de Londrina, 2013. Citado na página 5.
- MEFTAH, Mortharia et al. Ns-2 model of homeplug av plc technology. p. 231–239, 2011. Citado na página 5.
- MOHAGHEGHI, S; STOUPIS, J; WANG, Z. Communication protocols and networks for power systems-current status and future trends. p. 1–9, 2009. Citado na página 7.
- ORTEGA, Alcides. **Análise de desempenho de redes de comunicação wireless em aplicações de Smart Grid**. Dissertação (Mestrado) — Universidade Estadual Paulista (UNESP), 2015. Citado 9 vezes nas páginas 14, 31, 33, 35, 36, 37, 40, 58 e 70.
- ORTEGA, Alcides; SHINODA, Ailton Akira. Simulation in ns-2 of dnp3 protocol encapsulated over tcp/ip in smart grid applications. p. 1–8, 2013. Citado 5 vezes nas páginas 1, 2, 8, 54 e 55.
- PEREIRA, Cássia Correia da Silva. **Modelo de simulação NS-2 para o protocolo DNP3 sobre o protocolo de rede sem fio IEEE 802.15. 4 para simulação de baixo custo de aplicação Smart Grid**. Dissertação (Mestrado) — Universidade Estadual Paulista (UNESP), 2015. Citado 9 vezes nas páginas 2, 4, 8, 10, 14, 18, 30, 37 e 80.
- PEREIRA, Samuel C. **Estudo sobre a transmissão de dados via rede de energia elétrica em ambiente industrial através do padrão G3-PLC**. Dissertação (Mestrado) — Instituto Federal de Educação Ciência e Tecnologia de São Paulo (IFSP), 2014. Citado na página 46.
- PILARSKI, André Luís. **Power Line Communications (PLC)**. Dissertação (Mestrado) — Universidade Tecnológica Federal do Paraná, 2015. Citado na página 6.
- RAMALHO, LA et al. Modeling of state machines in vhdl for encapsulation of dnp3 protocol in p2p zigbee network. p. 1–8, 2013. Citado 3 vezes nas páginas 1, 7 e 26.
- ROSSI, Cristiane Zem. Plc: power line communications e sua projeção futura. Curitiba, 2013. Citado na página 25.
- SANTOS, Ana Fernanda Quaresma Batista. **Integração Matlab e NS-2 para um simulador de redes para o padrão DVB-S2**. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro, 2013. Citado na página 2.
- SANTOS, Christiane Borges et al. Modelagem de tráfego em redes plc (powerline communications) utilizando cadeias de markov. Universidade Federal de Goiás, 2009. Citado na página 4.
- SATO, Takuro et al. **Smart Grid Standards: Specifications, Requirements, and Technologies**. Chennai, India: John Wiley & Sons, 2015. Citado 4 vezes nas páginas 3, 20, 21 e 22.
- SCHREIBER, Jonas Fernando. **Modelagem de um Sistema de Distribuição de Energia Considerando a Aplicação em Redes Inteligentes (Smart Grids)**. Dissertação (Mestrado) — Universidade Regional do Noroeste do Estado do Rio Grande do Sul, 2013. Citado na página 5.

SHIN, In-Jae; EOM, Doo-Seop; SONG, Byung-Kwen. The coap-based m2m gateway for distribution automation system using dnp3. 0 in smart grid environment. p. 713–718, 2015. Citado 2 vezes nas páginas 2 e 7.

SILVA, Aislan Correia da. Estudo sobre a viabilidade da implantação da tecnologia plc. 2009. Citado 3 vezes nas páginas 24, 25 e 26.

SOARES, Pedro Hugo Anselmo. **Análise de soluções tecnológicas para comunicações Power Line Carrier (PLC)**. Tese (Doutorado) — Universidade do Porto, 2010. Citado na página 18.

TOLEDO, Fabio. **Desvendando as Redes Elétricas Inteligentes-Smart Grid Handbook**. São Paulo/SP: Brasport, 2012. Citado na página 3.

USMAN, Ahmad; SHAMI, Sajjad Haider. Evolution of communication technologies for smart grid applications. **Renewable and Sustainable Energy Reviews**, Elsevier, USA, p. 191 – 199, mar. 2013. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1364032112006107>>. Acesso em: 05 jun 2015. Citado na página 19.

VONDRASEK, Diane M. **The future os smart grid technology in the U.S. lodging industry: A delphi study**. 142 p. Dissertação (Master of Science in Hospitality Information Management) — Faculty of the University of Delaware, Delaware/USA, 2011. Citado na página 11.

WANG, Zhihai; CATABAY, Wilbur G; ZHAO, Joe W. **Process for treating damaged surfaces of low dielectric constant organo silicon oxide insulation material to inhibit moisture absorption**. Washington D.C./USA: Google Patents, 2000. US Patent 6,028,015. Citado na página 50.

XU, Shumin; YANG, Yatao. Protocols simulation and performance analysis in wireless network based on ns2. p. 638–641, 2011. Citado na página 8.

ZAMBENEDETTI, VC et al. Communication solution for implementing smart grid environment in the distribution network. p. 1–5, 2013. Citado na página 7.